
ACORN USER
For the BBC micro, Atom and Econet users
November 1982 £1
Simple machine code
Software for schools
Atom Graphics
Hints and tips
BBC news
TREK III
Space war game
page 15
WIN A BBC MICRO page 71
SOR

A bigger range than the Himalayas

The Acorn Atom From £118 plus VAT.

Personal Computing – Instructional and Fun

To get the best out of personal computing you need two things – hardware that is powerful and reliable – software that uses the hardware to the full. The Atom range is just that.

The Atom – tried and tested

The Atom was designed to last – inside and out. Outside a rugged, high impact case with a proper keyboard. Tested to withstand children as well as adults. Inside a powerful operating system that will never be bettered. It is available in several versions so you can choose what you want. And there is an enormous range of additional boards that fit inside the casing – start where you like – add more power, more versatility when you need it.

The Accessories – something for everyone

Diskpacks, printers, monitors, plug-in ROM's, manuals, other languages, arcade-type games, business and household software. Whatever you want to do – teach your children, run your business – you can't do better than choose the Atom range.

Available Nationwide

Not just mail order, the Atom range can be bought through a national dealer network – they will help and advise you. And in the unlikely event of breakdown they will be there. Like our equipment Acorn Computers are here to last.



Acorn Computers Limited,
Fulbourn Road, Cherry
Hinton, Cambridge CB1 4JN
Tel: (0223) 245200.

FREE Catalogue

For full details of the complete range and a list of dealers just fill in the coupon or write to us.

To: Acorn Computers Ltd,
Fulbourn Road, Cherry Hinton,
Cambridge CB1 4JN Tel: (0223) 245200.

Please rush me a complete list of the Atom range.

Name: ____________________

Address: ____________________

____

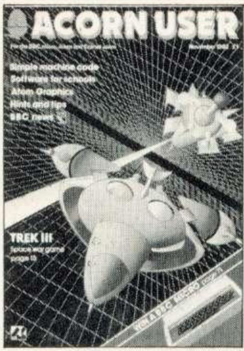
____

____

Postcode: ____________________

CONTENTS

ACORN USER Monthly
NOVEMBER 1982, NUMBER FOUR



Cover design by Chris Gilbert

How to submit articles
You are welcome to submit articles to the Editor of *Acorn User* for publication. *Acorn User* cannot undertake to return them unless a stamped addressed envelope is enclosed. Articles should be typed or computer written. Black and white photographs or transparencies are also appreciated. If submitting programs please send a cassette or disc. Listings should not contain more than 39 characters per line for ease of reproduction. Payment is £50 per page or pro rata. Please indicate if you have submitted your article elsewhere. Send articles, reviews and information to: The Editor, *Acorn User*, 53 Bedford Square, London WC1B 3DZ.

Coming soon in *Acorn User*:
● Sound on the Beeb ● Software reviews ● Atom word processing ● Maths in Basic ● Programs for schools ● Progress at the BBC ● Festive hints and tips ● Econet details ● Listings for games ● Teletext graphics

Editor
Tony Quinn
Managing Editor
Jane Fransella
Sub-editor
Ann Nimmo
Production
Susie Home
Marketing Manager
Paul Thompson
Promotion Manager
Pat Bitton
Editorial Secretary
Jane Lake
Publisher
Stanley Malcolm
Typesetters and Designers
GMGraphics
Graphic Designer
Phil Kanssen
Printed in Great Britain by E.T. Heron & Co. Ltd.

Editorial
01-631 1636
Advertising Agents
Dealer Deals Ltd
20 Orange Street
London WC2H 7ED
01-930 1612

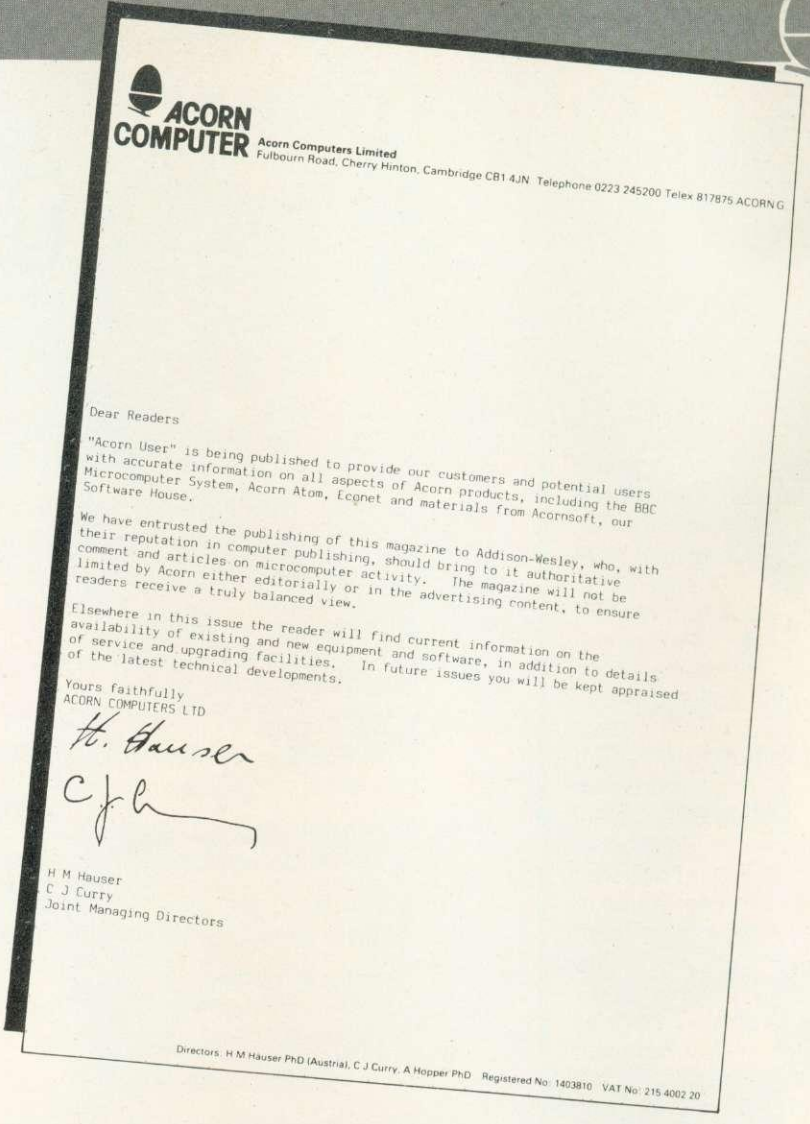
Published by
Addison-Wesley Publishers Limited
53 Bedford Square
London WC1B 3DZ
Telephone: 01-631 1636
Telex: 881 1948
ISSN: 201-17002 7

Subscription Information
For UK Subscriptions, send your cheque or postal order made payable to Addison-Wesley Publishers Ltd to:
MAGSUB (Subscription Services) Ltd, Ground Floor Post Room
Oakfield House
Perrymount Road
Haywards Heath
West Sussex RH16 3DH

Annual subscription rates

UK	*£15*
Europe	*£18*
Middle East	*£20*
The Americas and Africa	*£22*
Rest of the World	*£24*

Constructive approach to the problems of computing



ACORN COMPUTER Acorn Computers Limited
Fulbourn Road, Cherry Hinton, Cambridge CB1 4JN Telephone 0223 245200 Telex 817875 ACORN G

Dear Readers

"Acorn User" is being published to provide our customers and potential users with accurate information on all aspects of Acorn products, including the BBC Microcomputer System, Acorn Atom, Econet and materials from Acornsoft, our Software House.

We have entrusted the publishing of this magazine to Addison-Wesley, who, with their reputation in computer publishing, should bring to it authoritative comment and articles on microcomputer activity. The magazine will not be limited by Acorn either editorially or in the advertising content, to ensure readers receive a truly balanced view.

Elsewhere in this issue the reader will find current information on the availability of existing and new equipment and software, in addition to details of service and upgrading facilities. In future issues you will be kept appraised of the latest technical developments.

Yours faithfully
ACORN COMPUTERS LTD

H. Hauser
C J Curry

H M Hauser
C J Curry
Joint Managing Directors

Directors: H M Hauser PhD (Austria), C J Curry, A Hopper PhD Registered No: 1403810 VAT No: 215 4002 20

TO THOSE of you who have never seen a copy of *Acorn User* before – welcome. You are reading the only magazine for users of the Atom, BBC microcomputers and Econet systems which has official support from Acorn Computers.

Acorn User is relatively new – it was launched in July – and this is the first issue to be distributed nationwide. Its role is to help you squeeze the best from your micro (and help you get one), as well as providing news of the latest developments from Acorn, the BBC and any other sources which are relevant.

The word 'users' is meant in its widest sense as readers come from all worlds – education, business, industry, home – and many countries. So whatever your interest, there should be something here, if not let us know.

Commentators in the media have had a field day with their criticism of Acorn – much of it deserved. But this magazine aims to provide a constructive side to its criticisms. Feedback presented in a proper way is valuable both to writer and target.

This effect is appearing in Acorn's attitude to its new, mass market microcomputer – the Electron. Hopes of a Christmas launch have now been dashed, and the company is delaying the launch to avoid production difficulties which bedevilled the BBC machine.

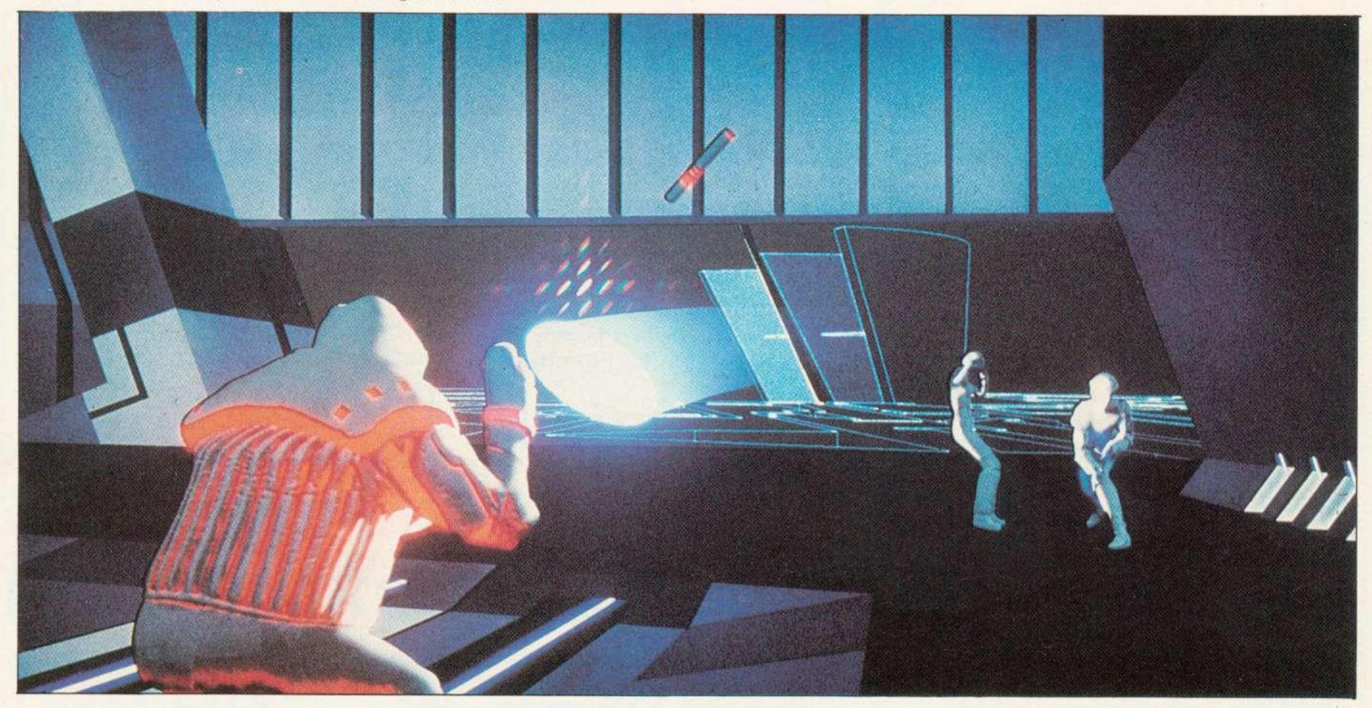
Anyway, less of all this theory and back to the interesting bits. David Allen is providing this month's article from the BBC on progress with the new TV series. But making programmes, just like building micros is all in the production, and serious filming won't start till early November.

Meanwhile, one of Auntie's offshoots, the Ceefax service, is preparing to launch telesoftware. Buying a receiver will enable computer users to download programs broadcast over the airwaves directly into the memory of their machines, and save them on cassette or disc. Imagine it, no more keying in! Prestel has also set up a computer database to provide a similar service, with the might of Buzby's British Telecom.

On the educational front, teachers are still faced with a dearth of software. As BBC micros spread through schools financed by the Government's microelectronics scheme, the need for decent programs will become more apparent. *Acorn User* hopes to remedy the situation by acting as a medium for ideas and printing listings.

That's all from the Editor for now, see you in December.





Acorn-Disney link up

British company joins forces with US film giant in schools competition

TRON, the science fiction adventure film, is the foundation of a major link up between Acorn Computers and American entertainment giant Walt Disney Productions.

A competition based on the film has been produced for primary and secondary schools in Britain – with BBC micros as the major prizes.

Acorn are offering a BBC microcomputer with disc drive, printer and software as first prize. There will be further awards given for regional entries.

But entrants need not already have a computer. All primary children need to do is describe the plot in 100 sentences of 10 words or less, while secondary pupils must write a story based on *Tron* laid out like a computer program.

Details are now being sent to schools.

At the risk of giving it all away, the story tells how Flynn, a programmer, has his video games stolen by a powerful computer company. In his attempts to find proof of the theft, Flynn is drawn into a computer.

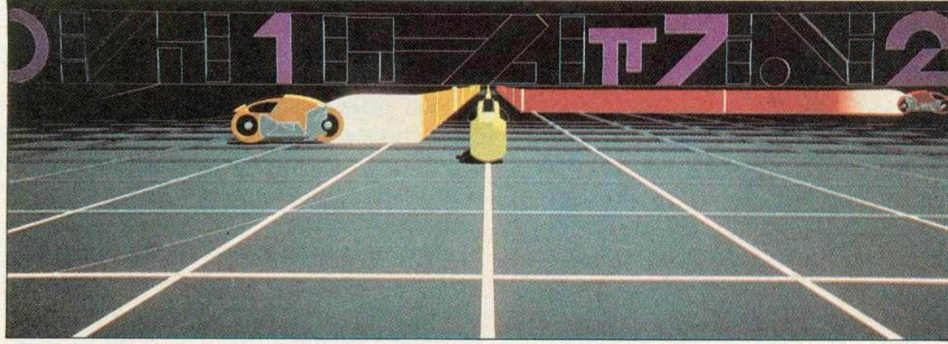
This fantastic electronic world is controlled by the Master Computer Programmer, who sentences Flynn to death by combat in the very video games he wrote in the outside world.

However, a mighty video warrior called 'Tron' comes to his aid in the battle against the MCP.

Much of the action takes place on light cycles, shown below. These leave a wall of energy behind them, which the riders use to try to trap their opponents. The machines travel at blinding speeds, and collision with one of these energy walls means death.

Tron lasts over 100 minutes, and makes extensive use of computer graphics. Disney spent over $20 million making the film – a quarter of this on the 15 minutes of wholly computer-generated sequences.

Unlike *Star Wars*, which used models and moving cameras to create battle scenes, *Tron* uses computers – with spectacular results. As the film's director of special effects said recently: 'It reminds you of something you have never seen before.'



Computer-drawn light cycles play a deadly game



Electron launch delayed

THE Electron will not be available before the end of the year – that's the disappointing news from Acorn.

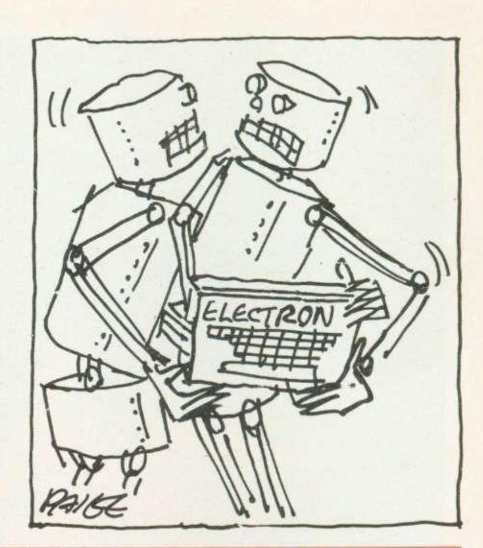
The company had hoped for a Christmas launch and the machine was on schedule a month ago – with just the ULA to be finalised. However, this chip was always the determining factor, as director Chris Curry said in last month's *Acorn User*.

The ULA is not ready for main production, and so the project is being re-scheduled for spring.

Acorn's £150 Electron is designed to compete with the Spectrum, and will use add-on modules to provide interfaces for Econet, teletext and printers.

Another project still in the pipeline is a microdrive. The first hints of this 3½" micro-floppy were seen in *Acorn User*, and Acorn are still evaluating prototypes.

Meanwhile, games paddles at £13 a pair are being despatched and production of the voice synthesiser ROM is underway.



Australian inroads

BBC micros are beginning to make inroads into foreign markets.

The machines have already been accepted for schools in three Australian states, Tasmania, Victoria and Western Australia.

Progress has been slow as output has been fully taken up with supplying demand in Britain.

The BBC plans to export is *Computer Programme* to countries which have shown an interest, such as Australia.

Reports in Australian newspapers claim the country has more networks in schools for Beeb machines than England. Britain is claimed to have just two!

Beeb teletext expansion

TELETEXT adaptors for the BBC microcomputer system are now in production.

These will give the computer the ability to decode Ceefax transmissions, and download and save software from the BBC. Hence users will not have to key in programs.

The receiver will cost £225, including VAT. It comes in a case about the size of BBC micro cut in half from front to back, and will be made from matching plastic.

With each adaptor, a ROM is supplied to be fitted to the model B which contains the interface software to make the computer operate as a convential Teletext receiver. It also allows the computer to receive programmes which may be loaded and run in the computer, ie telesoftware. This chip will be fitted by dealers.

Within the case are contained a television tuner section, digital circuitry for teletext acquisition and a power supply. There is a mains lead and TV aerial connector on the rear panel and underneath the front of the case a cable joins the 1MHz bus connector on the model B.

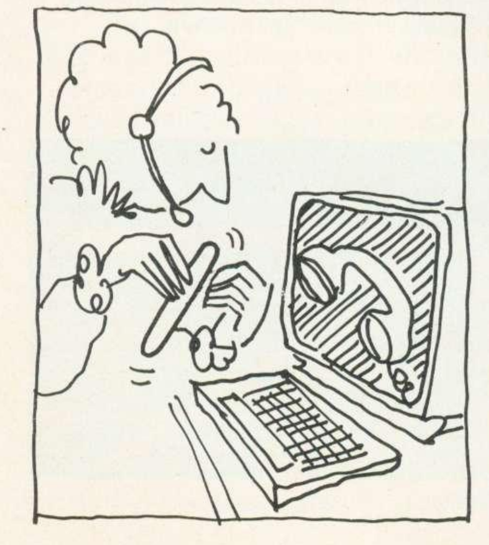
On the rear panel, four controls each offer a tuning range over the UHF TV channels E21 to E69 (470MHz to 860MHz). A tuning aid program in the teletext software gives a graphic representation of tuning condition when setting up these four channels. Once adjusted at a particular user site no further tuning is required, allowing selection from four pre-set frequencies.

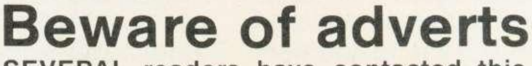
For good reception, an external UHF TV aerial which is not fed to the adaptor via a cable distribution system is normally required. The computer is designed to guarantee a teletext decoding margin in excess of 70% (50% would normally be sufficient).

Digital circuitry within the adaptor receives TV lines of teletext information every field period. Up to eight lines (320 characters), may be received in each period, although only two lines of teletext per field are being broadcast at present. In the UK with a TV field frequency of 50Hz, this means that the adaptor has an input data rate of 128k bits per second. (There is provision within the hardware for extension to 16 lines.)

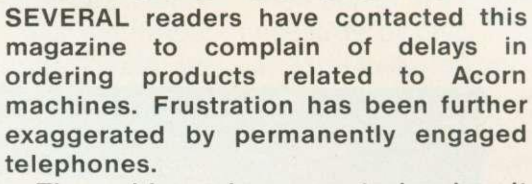
After the teletext lines in each field, the adaptor interrupts the BBC micro and passes the teletext data received to the computer. If these lines are required to build up teletext page the computer is trying to acquire, they are stored, otherwise they are ignored.

The BBC microcomputer contains a full implementation of the teletext, including upper and lower case characters, colours, separated and contiguous graphics, double height and flashing characters.





Beware of adverts



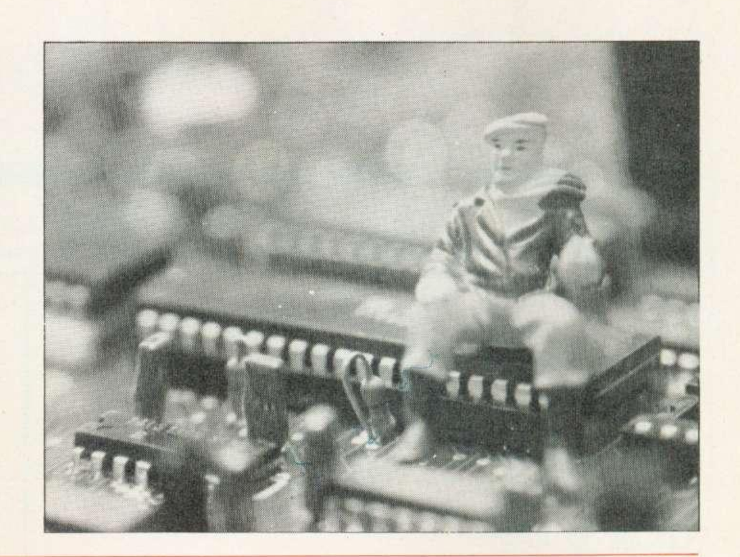
SEVERAL readers have contacted this magazine to complain of delays in ordering products related to Acorn machines. Frustration has been further exaggerated by permanently engaged telephones.

The golden rule seems to be: buy it over the counter, or check it really is in stock before ordering by post. Reputable companies will be pleased to provide advice.

One company has been advertising BBC teletext and Prestel receivers as 'in stock' since September. In fact, the first is only just becoming available and the second isn't even in production.



Shrinking micros



THIS little chap is straight from a toy sports car. But what's he doing inside a microcomputer? That question will be answered by BBC TV's new computer series, which begins filming in November.

The plan, apparently, is that Ian McNaught Davis – 'Mac' – is going to do for the inside of computers what David Bellamy did for back gardens.

It is just one of many ideas being tried out for the new programmes which will be shown in the New Year.

David Allen, the producer, gives the lowdown on some of the plans he has on pages 10 and 11. Meanwhile, you'll just have to be content with repeats of the *Computer Programme* on BBC1, Sundays and Mondays.

Black box hooks up to robots

GIVING micros power over robots is the role of the black box pictured here.

The Bedfordshire Interface Module is designed to drive hydraulic and pneumatic valves, motors, and solenoids used in robotics.

Any micro with an eight-bit user port is suitable for the interface, which has served a year's apprenticeship in a mobile classroom.

Bedfordshire Education Authority's Technology Bus, (part of an A-level technology course), provided the test-bed.

Several other modules plug into the black box to provide joysticks, digital to analogue conversion or a speaker box.



Input to the module can be from four push-buttons on the front panel, or from microswitches, thermostats etc, connected via a 5-pin DIN socket or 3.5mm jack sockets.

The input signals go via a Schmitt trigger circuit to remove any noise and then on to opto-isolators. This ensures the computer input port is not damaged by inadvertant connection to high voltages.

Its makers, Educational Electronics can provide more information:
30 Lake Street,
Leighton Buzzard,
Bedfordshire LU7 8RX.

Network brains ring the changes

ORBIS, the brains behind the Cambridge Ring have joined forces with Seel of Edinburgh to produce local area networks.

The Edinburgh company will manufacture and distribute Orbis products, including the 16-bit network processors, based on Motorola's 68000 chip.

Co-operation between the two has already resulted in successful installations at Cambridge University, Rutherford and Appleton Laboratories, British Telecom Research and ICI.

Under the agreement, 68000 network systems will be supplied with Mace high speed intelligent network interfaces, providing on-board protocol software, up to transport layer (level 4). A range of Mace host adaptors, initially to Vax, PDP-11, Prime and GEC 4000 machines are also under development.

For their part, Seel will make available their range of interrupt-driven interfaces eg Multibus, Q-Bus and S100 together with network nodes, conforming to the new CR82 British standard.

Orbis is part of Acorn Computers, and details of the network systems are available from Peter O'Keefe on (0223) 312449.

Dealer list

THERE are several changes to the dealer list on page 72, which we were unable to include.

The Microstore, London SW3 is not answering its phone and appears to have closed down.

Also the Typewriter centres in the Midlands are no longer Acorn dealers.

Micro knick-knacks

ALL sorts of weird and wonderful things are popping up to add to your Acorn micros.

Computer graphic design sheets (£4.95 for 100-sheet pad), decimal/hex scale rules (£7.95) and TV support stands (£29.50) are three of the latest products.

All are produced by Dracal Design Consultants, who will give bulk discounts to schools and members of some user groups.

The graphic design pads come with photostated sheets giving details of setting up user-definable characters, with practical examples of producing aliens and UFOs.

Information from Dracal (North West) Ltd, PO Box 130, Warrington, WA1 4QB



Modelling aids for education

A RANGE of products which model engineering devices has been produced for the Beeb and Atom.

These modules mimic equipment such as washing machines or diesel generators, to give the user some idea of control and feedback techniques.

The range, called Mica, is made by Feedback Instruments Ltd and designed as an aid in teaching robotics, environmental engineering and numerical control.

Applications modelled include: washing machine; stepping motor; diesel generator; temerature control; traffic control; binary input/output.

There is also an electronics project board and multifunction input/output available.

Feedback Instruments also produce interfaces for the application modules. The company's address is: Park Road, Crowborough, Sussex TN6 2QR.

Acorns overseas

SEVERAL readers have asked about distribution of Acorn computers overseas. The following companies should be able to give details:

AUSTRALIA
Consolidated Marketing Corporation (Imports) Ltd
Melbourne 419 3033

BELGIUM
Computing & Electronics International NV
Antwerpen 32 08 09

DENMARK
Bergqvist & Hobberstad Eng. A/S
Copenhagen (451) 133188

FRANCE
J C S Composants Sarl
Paris 1 355 9622

HOLLAND
Compac/Acoustical BV
Kortenhoef (35) 61614

IRELAND
Lendac Data Systems Ltd
Dublin 372 052

ITALY
Iret Informatica
0522 32643/4/5/6

PORTUGAL
Datamatic
Braga 71555

RSA
Durban 325 531

SWEDEN
Beckman Innovation AB
Stockholm 390 400

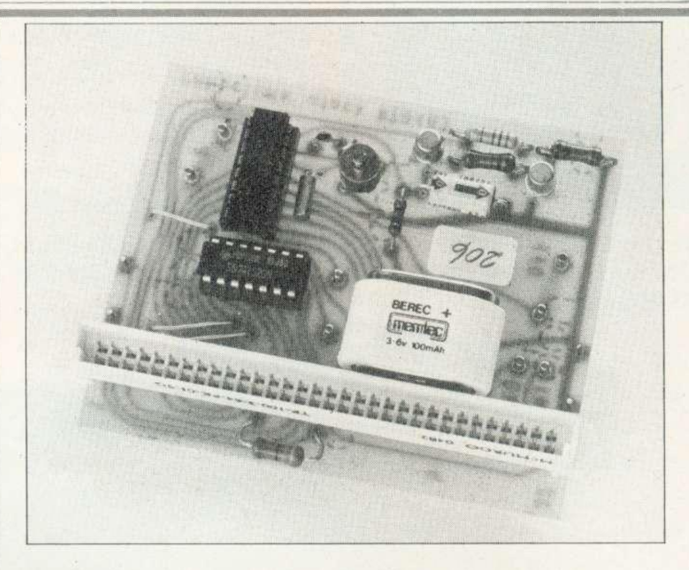
Free advice

INDUSTRIAL and business readers in the Sheffield area can lay their hands on a micro at an advice centre in the city.

The South Yorkshire Microsystems Centre will be run by Sheffield Polytechnic, and forms part of a network set up by the National Computing Centre.

Advice will be free, but more complex consultancy sessions will be charged for.

The address is Dyson House, Suffolk Rd, Sheffield S1 1WB. Tel: 0742 738621



Time for the Atom

FOR those Atomists to whom time is of the essence, Varuna Electronics has produced a real-time clock and calendar.

The board fits inside the Atom's casing, and comes complete with its own battery for power when the machine is switched off.

Time and date are always available, and leap year calculation is automatic. Accuracy is claimed to be within one second a day at 20°C.

At £20 (inc postage and VAT), the clock comes with five sample programs and full instructions. Contact Varuna Electronics, Horsell Park, Woking, Surrey GU21 4LY.

Characters

USERS with an Okidata Microline M80 matrix printer can now buy a character generator EPROM which matches Atom and Beeb keyboard codes.

The generator has ASCII special symbols, slashed and unslashed zeros and costs £15 (plus VAT).

Computech, of 168 Finchley Road, London NW3 6HP, produce the firmware. They will supply the M80 and generator for £230 (plus VAT).

Beeb case

IF carrying your micro is a weight on your mind, special cases are available.

W.H. Hayden supply steel edged fibreboard cases at £34.95 (excluding postage and VAT). Details from 52 Holloway Road, London N7 8JL.

Meetings focus on microcomputers

■ Microfest 82. December 11, 12. Repeat of exhibition held earlier in the year. Venue is UMIST in Manchester. Details from Richard Hewitt, Third floor, 121 Princess St, Manchester M1 7AG.

■ Development of new Teaching and Learning Methodologies, December 12-17, Bristol. Sessions on the BBC micro, December 12, 14. Fee is £82.50 including accommodation. Details from Registrar, Further Education Staff College, Coombe Lodge, Blagdon, Bristol BS18 6RG.

■ 'Can we use a desktop here?' Symposium of Institute of Chemical Engineers and Society of Chemical Industry on use of personal computers in the process industries. Contact either organisation.

■ Expert Systems Tutorial, December 20, London. Fee is £40. Details from Richard Forsyth, Maths Dept. Polytechnic of North London, Holloway Road, London N7



ROBOTS GO SHOPPING

David Allen has stepped in to Paul Kriwaczech's shoes to produce the second BBC computer series. Here he details the programme's progress, and some of the strange things that go on

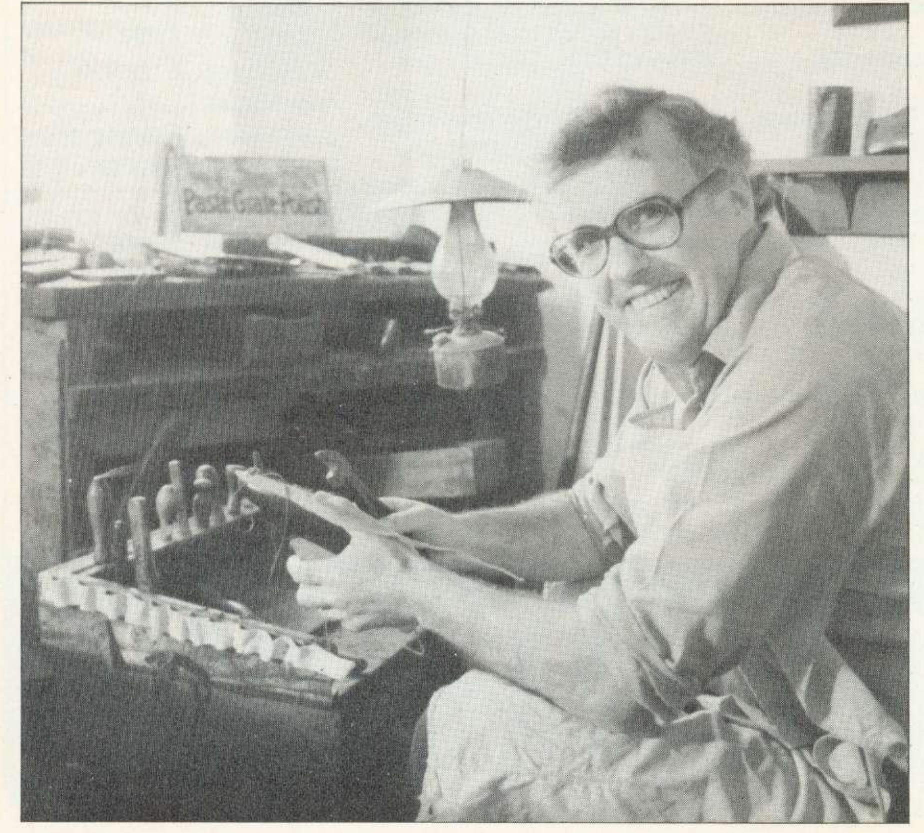
If you're watching the latest repeats of the *Computer Programme* (Sunday mornings on BBC1) you'll maybe be interested to know how next year's series – *Making the Most of the Micro* – will build on what has gone before.

Originally, we conceived the *Computer Programme* as a 'hands-on' series concerned with 'how to do it'. Later we changed the philosophy because it became clear that what was needed was a series designed to explain the basic ideas of the wide world of computing to the general public – especially those with no experience of computing at all.

My colleague, Paul Kirwaczeck, achieved that elegantly in series one; it's now my job to produce a series for micro-owners and the technically curious – a heterogeneous audience, but including those who, perhaps thought series one was simple and slow.

Of course, we've got to be careful: our ultimate business is to illuminate, stimulate and inform our audience. But that audience does not just consist of BBC microcomputer owners. Whatever machine you might have, and however much knowledge you have, the television series has got to appeal and have some relevance – even if it is simply reinforcing ideas which you already understand – perhaps explaining them in a new way. This has forced us to look hard at the fundamentals of using micros – at the common ground of personal computing.

Cobblers . . .'Mac' shoes how it's done



Take writing programs in Basic. Many viewers will not want to know how to do this, and others will already have a much greater knowledge than we can put over. Nevertheless, we are going to show Ian McNaught Davis – 'Mac' – doing some coding in most of the programmes.

What matters to us and our advisors, in Programme 2, for example, is how to get across the idea of the main programming structures – that, amid all those parochial, machine-dependent techniques, there are only a handful of principles being applied. In showing this, BBC Basic is a great strength: long variable names, IF . . . THEN . . . ELSE, REPEAT . . . UNTIL and procedures make it so much easier to make a program clear to read – and that is important when we are trying to show these things on tlevision.

If you are working on a machine with Atom Basic, Sinclair Basic or Pet Basic, the code will be different, but the programming concepts – which is what really matters – will be just the same. On the other hand it would be foolish to give details of how to format the screen on a particular machine – each is different. But it is useful to look at good presentation – for example, when to avoid the top line of the display, being consistent about the centering or aligning of lines, and so on.

As before, the use of analogy is important as a way to get ideas home. So, in Programme 2, writing a program in Basic is likened to ordering a meal in a Chinese restaurant. The order must be given in 'appropriate' language – and then translated into hieroglyphics


Waiter . . . is there any machine code on the menu?

by the waiter (microcode) which the kitchen staff act on using their own local rules unbeknown to the customer (system software) – unless, of course, he wants to run his own Chinese restaurant (eg write in machine code).

Later in the series, we examine how to write a long program. This nearly always involves breaking things down into manageable chunks by the use of subroutines or procedures. So Mac is seen in Programme 3 in a supermarket, speculating about how a robot might be sent on a shopping spree, with orders to invoke PROC BEST-BUY, instructions on how to buy avocados and tomatoes with PROC PICK'N'WEIGH and elaborate details for PROC CHECKOUT.

Don't be alarmed at the thought of all that code – there'll be plenty of other things in the series. We'll be looking at the use of a home database, the fundamentals of which are introduced by Mac in the BBC's gramophone library with its one million records. (No, we don't suggest a micro could do that job!)

We'll look at word processing, artificial intelligence programs, at computer-aided design and so forth. And – important to our overall aim – most of the software you see in the series (like the earlier series) will be available in some form or another for BBC micro-owners to get hold of, primarily through BBC Publications.

Applications of all kinds will be shown – on a range of micros. Here we're only interested in what's behind the applications – the hardware is of secondary importance. And when we stray from the strict field of the micro (for example when we look at computer-aided design of shoes at Clarks factory in Somerset), the aim will be to see how the humbler machines can do similar – if more limited – things. In this case, manipulating a three dimensional 'model' with a joystick, introducing hidden line removal and allowing the user to colour the object in – not with one of a million different possible colours (as with the shoes) but with about four.

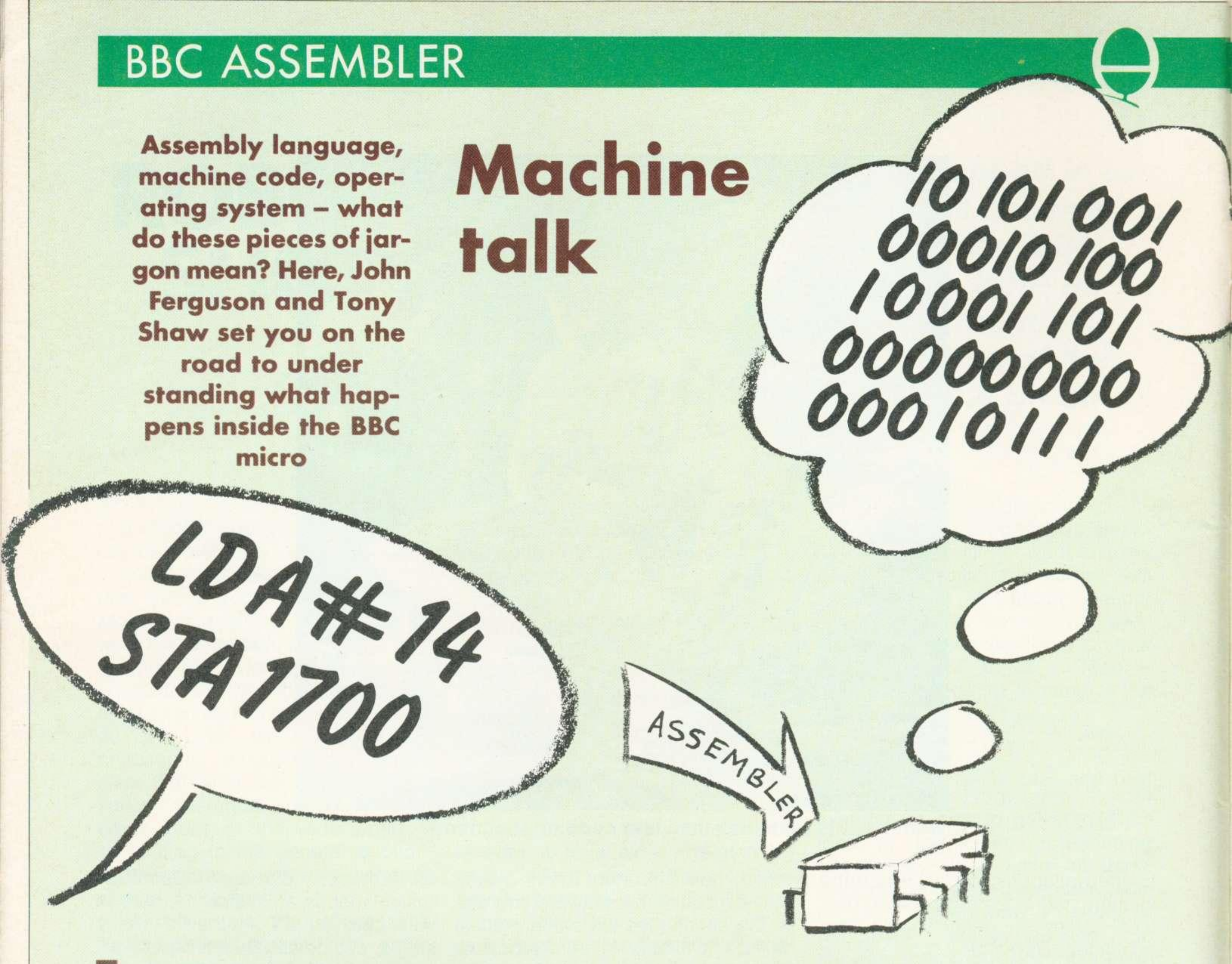
Our examples must show the limitations of the technology as well as show its strengths – so here the relatively slow speed of the micro will be shown up (to give the lie to all those clever, apparently instantaneous animation sequences you see on TV – they're really single, computer-generated pictures, shot one frame at a time on old technology (ie film). Finally, in that programme, we hope to show a professional two-dimensional CAD package (soon to be available on the BBC microcomputer), using the second processor and a precision joystick.

In other parts of the series there'll be demonstrations of the use of speech and music synthesisers, of how to get various sensors and control devices interfaced to the micro and how to use it as a communications device. For a rough breakdown of the subjects of each of the ten programmes, see *Acorn User's* July issue.

There's an old adage in broadcasting – 'never work with children or animals'. We have now expanded it to include micro-electronics! As I write, we have yet to record a single programme: watch this space for news of our progress – above all, watch the series. Fingers (if not wires) crossed, it starts in January, late Sunday evenings, BBC1.

Assembly language, machine code, operating system – what do these pieces of jargon mean? Here, John Ferguson and Tony Shaw set you on the road to under standing what happens inside the BBC micro

Machine talk



Following in the footsteps of the Acorn Atom, the BBC micro contains a machine code assembler 'embedded' in the Basic interpreter. The version on the Beeb is even friendlier, providing the ideal tool for both the novice and the expert to tackle assembly language programming.

The beginner might ask: 'Why bother with assembly language?' The classic reply is that the microprocessor's own language allows you to tackle problems where speed is important, where actions are required in microseconds rather than milliseconds, or where memory space is restricted.

Assembly language also allows you to 'lift the lid' and dip into the machine operating system, the machine code program that runs the Beeb.

The language used by the microprocessor bears little resemblance to Basic. The processor follows instead a numeric language of 0's and 1's called machine code. For example, the set of binary numbers in the following program instructs the microprocessor to store the hex number 14 in memory location 1700 (hex).

Binary	Hex representation
1010 1001	A9
0001 0100	14
1000 1101	8D
0000 0000	00
0001 0111	17

A machine code program is similar to a Basic program in that both are a mixture of instructions and data. In the above example the code A9 tells the processor to load its accumulator (an internal register) with the next number, 14. The next code, 8D, instructs it to store the contents of the accumulator in the memory location defined by the next two numbers ie in 1700. The numbers representing the instructions are called operation codes or 'op codes'.

I is possible to write programs directly in machine code, but the process is slow and prone to error, requiring continuous reference to tables of operation codes (*User Guide*, p508).

An alternative approach is to write programs in a more human and friendly format called 'assembly language' where alphabetic abbreviations, rather than binary or hex codes, are used to represent the instructions.

Abbreviations LDA and STA, for example, are used to represent the operations 'LoaD the ACcumulator' and 'STore the Accumulator'. These mnemonic abbreviations are much easier to remember than op codes.

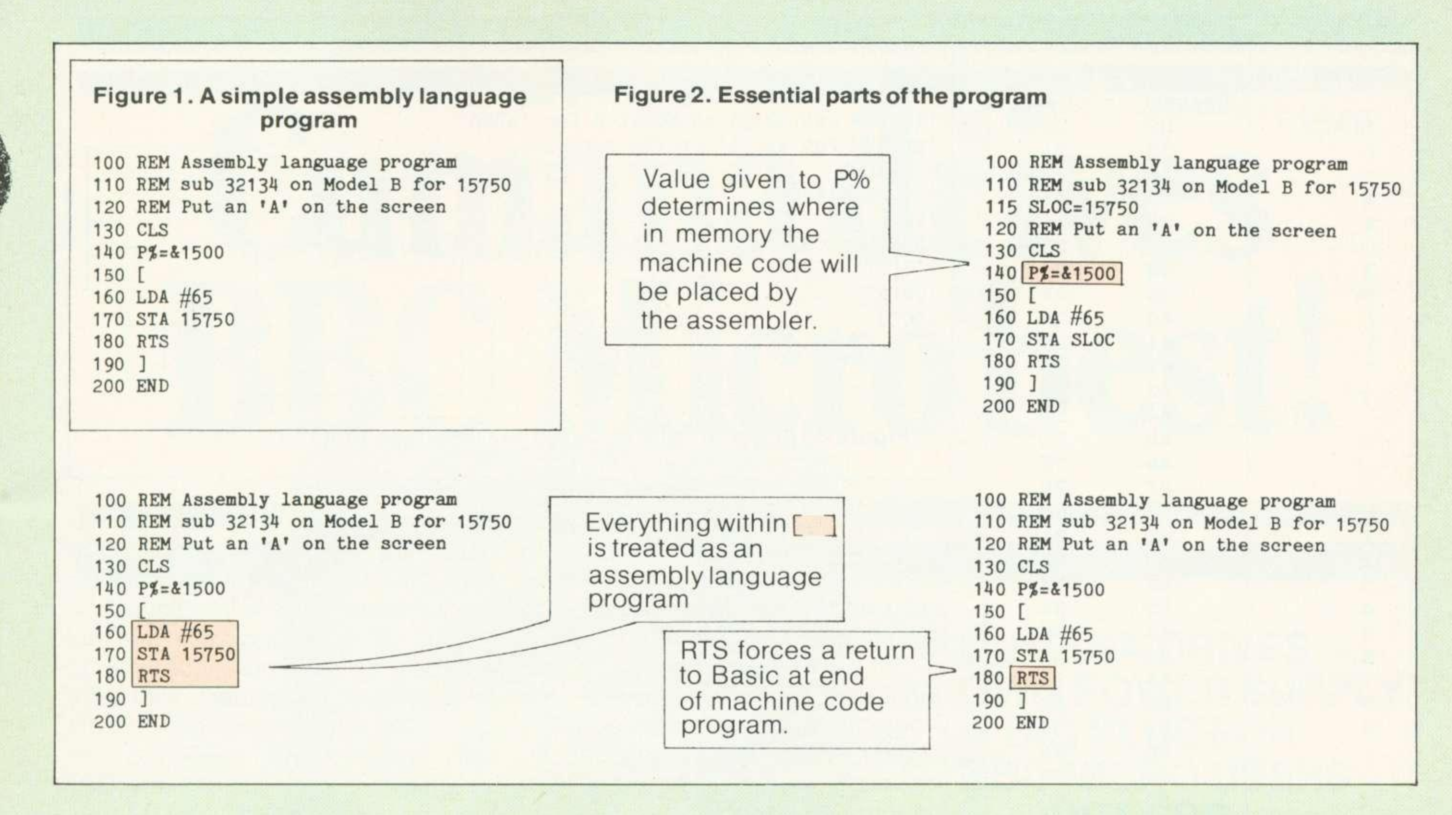
Written in mnemonics the above example becomes:

```
LDA #14
STA 1700
```

But how does this assembly language program become a machine code program? The answer is to use a special translating program called an assembler, which translates the 'friendly' mnemonics into machine code (figure 1).

You do not have to be an experienced 6502 asembly language





programmer to be able to use the BBC assembler. The following simple program can be entered, assembled and run by someone with no previous knowledge of assembly language.

Using the BBC computer, there are three essential features of any assembly language program (figure 2).

The most fundamental is the pair of square brackets – [ ] – (arrows in teletext mode). These tell the Basic interpreter that what appears between the brackets is to be treated as an assembly language program.

The value given to the parameter P% determines where in the memory the machine code program will be placed by the assembler.

The 6502 assembly language instruction RTS placed as the last item within the assembly language program ensures that program control will ReTurn to the Basic interpreter.

In the program shown in figure 2:

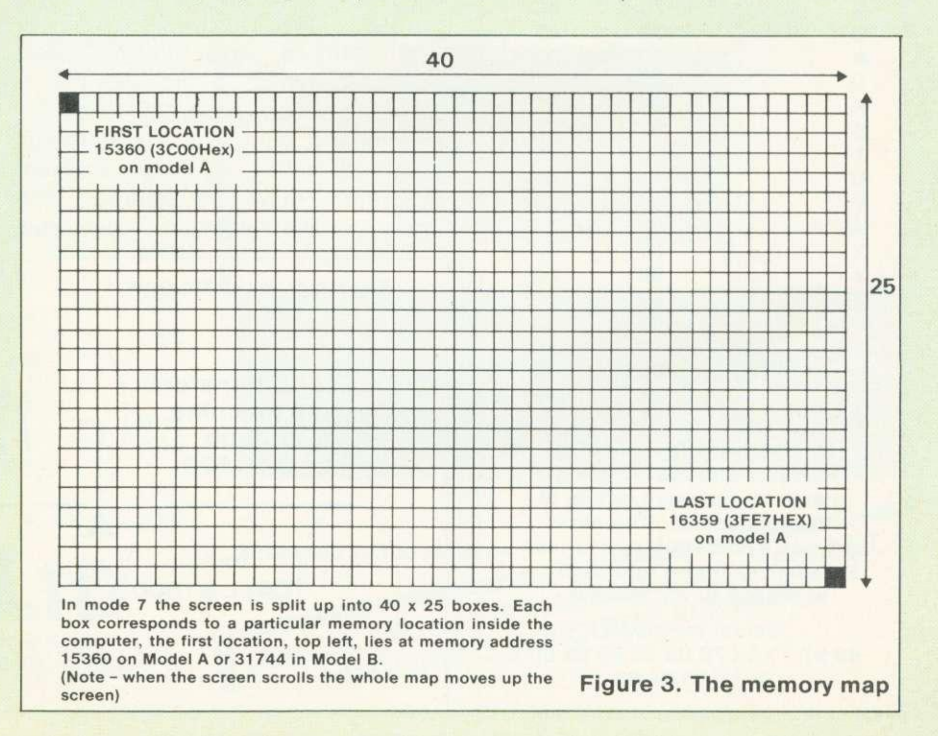
LDA #65 loads the accumulator with the value 65.
STA 15750 stores the contents of the accumulator (65) in location 15750.

When in mode 7, this will make an 'A' appear on the screen, since location 15750 is part of the memory-mapped screen display and 65 is the ASCII value for 'A'. (Note that model B users should use location value 32134.) The phrase 'memory mapped' and the abbreviation 'ASCII' are briefly explained in figures 3 and 4.

Before attempting to assemble the program, ensure your computer is in mode 7. The following sequence of commands will then make our simple program perform its intended task.

```
>RUN
>CALL &1500
```

Each of these commands performs a function essential to the successful operation of the program. In running the Basic program, the assembly language statements contained between the square brackets are translated into machine code. The translated program – essentially a series of 0's and 1's, is placed into an area of



In mode 7 the screen is split up into 40 x 25 boxes. Each box corresponds to a particular memory location inside the computer, the first location, top left, lies at memory address 15360 on Model A or 31744 in Model B.
(Note – when the screen scrolls the whole map moves up the screen)

Figure 3. The memory map



CHARACTER	CODE Decimal	Hex
SPACE	32	20
!	33	21
“	34	22
#	35	23
$	36	24
%	37	25
&	38	26
⅜	39	27
(	40	28
)	41	29
*	42	2A
+	43	2B
,	44	2C
–	45	2D
.	46	2E
/	47	2F
0	48	30
1	49	31
2	50	32
3	51	33
4	52	34
5	53	35
6	54	36
7	55	37
8	56	38
9	57	39
:	58	3A
;	59	3B
<	60	3C
=	61	3D
>	62	3E
?	63	3F
@	64	40
A	65	41
B	66	42
C	67	43
D	68	44
E	69	45
F	70	46
G	71	47
H	72	48
I	73	49
J	74	4A
K	75	4B
L	76	4C
M	77	4D
N	78	4E
O	79	4F
P	80	50
Q	81	51
R	82	52
S	83	53
T	84	54
U	85	55
V	86	56
W	87	57
X	88	58
Y	89	59
Z	90	5A
←	91	5B
½	92	5C
→	93	5D
↑	94	5E
▛	95	5F

Computers work only with numbers and use a coding system to represent letters of the alphabet etc. The most common system is the American Standard Code for information exchange or ASCII code.

Secret message
65 67 79 82 78 83 32 65 82 69 32 78 85 84 83

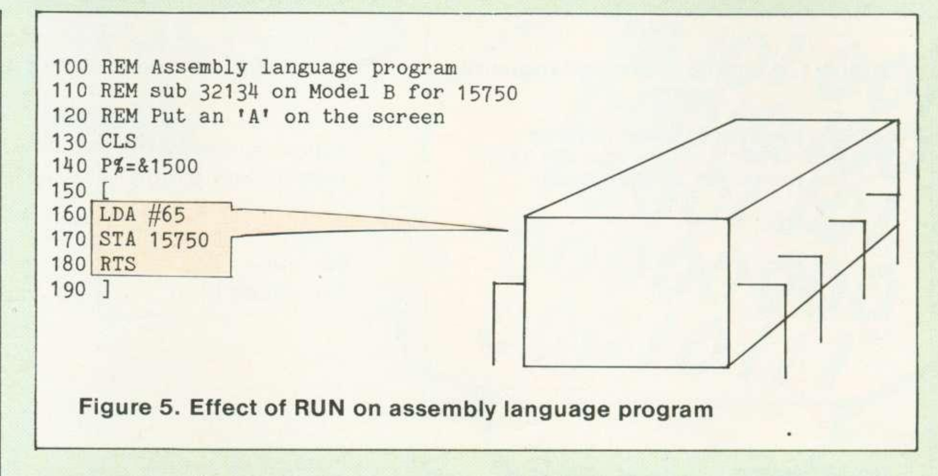


Figure 5. Effect of RUN on assembly language program

memory beginning at P% – &1500 in our example (figure 5). A listing will also be produced, which will be described later. It is possible to turn off the listing function.

Call is a Basic command that transfers program control to the program residing in memory at the location specified following Call. So this command is required to run our program, which was placed in the locations following &1500. The effect of CALL is illustrated in figure 6.

Once control has been transferred to the program, starting at location &1500, the computer's microprocessor will automatically attempt to execute the contents of succeeding memory locations. The important RTS is executed after the two instructions of our program (LDA and STA) have been carried out. This forces program control to return to the Basic interpreter – and normality is restored.

So the BBC assembler has been successfully used to create a machine code program from the assembly language statements included within our original Basic program. There are features that make the BBC especially powerful, and some of these will be illustrated later. But to continue with the simple example, what do the contents of the listing tell us?

The listing displays the memory address and memory contents in hexadecimal, together with the corresponding mnemonics. RUN will automatically generate an assembler listing, unless an OPT statement (*User Guide* p314) has been used.

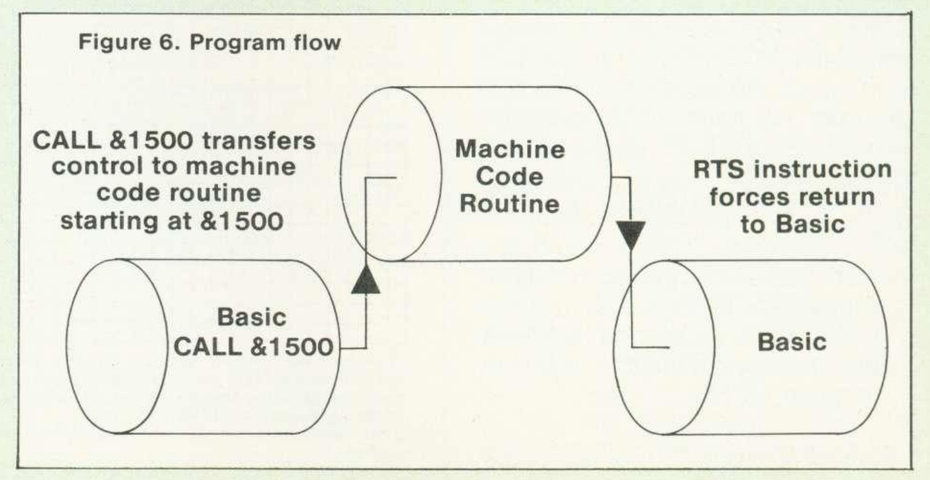


In the same way that the Rem statement can be used to include comments within a Basic program, comments can be incorporated in an assembly language program by preceding them with a backslash character (/). In the teletext mode, this is displayed as one half (½).

Comments are an important part of any program. However, given the lack of 'readability' of assembly

▶page 78

Figure 6. Program flow





PROGRAMMER'S FORUM

The chance for readers to earn up to £20 by providing original hints and tips. Coping with BREAK is the subject of this month's column

THIS is the start of a new column in *Acorn User.* In the next three issues, Ian Birnbaum will be providing ideas on increasing the programmer's control and understanding of the BBC micro. This month he looks at the BREAK key. In the next two months, among other things, he will be showing you how to cope when you type OLD or O. instead of OLD or O.; how to save your program from within the program itself. But the real purpose of this column is for you, the readers, to send in your own tips.

These three columns should give you an idea of the sort of thing we want. What matters is that the hint is original, in that the gist of it should not have been published in any other magazine, and that it is described clearly and fully.

We will pay £5 for any hint published, with £10 for any which merits a one-star award, and, exceptionally, £20 for any which merits a two-star award.

Hints should be typed or computer written, and any substantial listings should be on cassette (programs should be included only if they illustrate a point). Mark the top of your contribution 'BBC Forum'. If you want your contribution returned, please enclose a stamped addressed envelope.

This column is also here to solve programming problems. If you have a problem you want answered, mark your letter 'Problems'. Ian cannot answer letters personally, but will endeavour to include a cross-section of the most interesting and commonly-occurring issues.

YOU can program the BREAK key *KEY 10, and this can be used to effectively 'disable' the Break key within a program.

```
*KEY10 OLD|MRUN|M or
*KEY10 OLD|MG.100|M
```

are the usual ways of doing this. In this sense, you are trapping BREAK in the same way you trap ESCAPE with ON ERROR.

But there is a difference. With ESCAPE you do not lose all the values of the variables within a program: with BREAK you do. This can be important in word processing, for example, where hours of editing and appending text can be wasted by the simple accident of pressing the BREAK key. However, there is a way to restore the values lost, and to build this into the programming of the BREAK key.

To understand how this is possible we must first understand how the BBC micro records the values it uses. First, the so called static variables, A% to Z%, are stored in fixed memory from &404 to &46B and are unaffected by the BREAK key (@% is also static, stored at &400 to &403, but this is affected by the BREAK key). Therefore if you use static variables for all your integer values BREAK will not harm them. This has the disadvantage of reducing readability in your program, but there is a second advantage of their being faster to access than variables with long names.

Location

All string, floating point and non-static variables are dynamic in that



Conducted by Ian Birnbaum

This column will also answer readers' queries. Hints or problems should be addressed to Beeb Forum Acorn User, 53 Bedford Square London WC1 B 3DZ

```
10 A=5:B=9:C=4:D=9:T=7:@%=4
20 PROCTEST
30 !&DF0=!2+24:!&DF2=!&482:!&DF6=!&48
   6:!&DFA=!&4A8
40 *KEY10 O.|M@%=4:?2=?&DF0:?3=?&DF1:
   !&482=!&DF2:!&486=!&DF6:!&4A8=!&DFA|MG.
60 |M
50 A1=6:A2=1:A%=GET
60 E=3:PROCTEST:PRINTA,A1,A2,B,C,D,E,
   T:END
70 DEF PROCTEST:PRINT"OK":ENDPROC
```

their position in memory depends upon the program in which they are used and on the order they are accessed within that program. The location of the first variable of each initial letter is stored in a fixed place, however. The location of the first variable starting with A which has been met in a program is stored at &482, that with B at &484 and so on up to Z at &4F4.

When you type RUN, all these locations are set to zero (ie each contains &0000) : this is also true after you have pressed BREAK. It follows that if we can store the contents of these locations prior to BREAK being pressed, then we can restore them after BREAK has been pressed.

Since we do not know when BREAK will be pressed, this means we will have to initialise our variables early on in the program before we put the relevant details into memory. This has already been suggested for strings (see article on string handling last month page 24), so it means that in addition it has to be done for floating point and non-static integers. This could be very laborious in some cases, but there is a way to make it easier. If, for example,all your floating point and integer variables begin with the letter A then you need only initialise one of them early on. The rest are fixed relative to this first one in that the place where the first is kept contains the address where the second is kept, and so on.

Speed factor

Of course, it is rather extreme to have all your variables starting with A; it also slows things down since the interpreter will have to skip through quite a few variables in memory to get to the A's at the end of the list. But if you always arrange to initialise first those variables where speed is a factor, and to use only a few of the available letters to start your variables, then it is not too laborious to initialise the key variables prior to saving locations in memory.

One more fact is required before we see how to program the BREAK key. The interpreter has to keep track of where the first free location in memory is at any time: the information is kept in locations two and three. This information will change as more variables are created. For this reason, we must reserve space for all the variables uninitialised prior to the saving of two and three. As a rule of thumb, allow 10 plus the name length for each integer or floating point variable. Also dimension all arrays prior to saving two and three.
We are ready to look at an example now. Run the program given.

When OK comes up, press BREAK – the values of all the variables will be displayed. Moreover, the PROC will still work: the reason for this will be discussed in next month's column.

Contamination

It should be clear how this program works. Line 30 saves the relevant values (using the 'pling' operator – see chapter 39 of the *User Guide* for details) in &DF0 to &DFB: these are otherwise unused in a standard system. Line 40 programs the BREAK key to put these saved values back into their original places. We cannot generally use !2 here for it would contaminate locations four and five which is the stack pointer (in fact it would be safe in this particular example).

Experiment with this, and try to incorporate the idea into your own programs where pressing the BREAK key could otherwise be disasterous.



In a far-flung galaxy on an uncharted edge of Earth's decaying Empire, a lone Starship fights a losing battle as it stalks its prey – the Klingons.

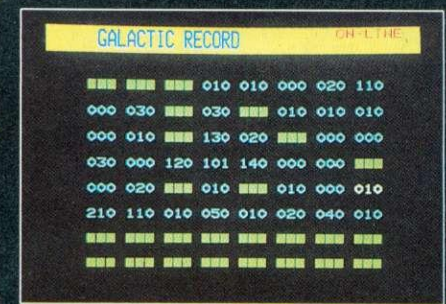
But the enemy ships are lying dormant, waiting for the Earthship to come within range. Then, they pounce. And the battle-weary humans can only take so much. Their precious energy dwindles with every move and each encounter saps the Starship's strength.

But the Klingons can be beaten. Supplies are available from Starbases – though they are few and far between, and their positions unknown. So it will take careful planning and every last ounce of the Starship Commander's cunning to win.



TREK III

A STARSHIP'S LAST STAND



Your job as Starship Commander is to destroy all 15 Klingons (red arrows), using as little time and energy as possible. Your initial energy level dictates the level of difficulty.

First you have to find the enemy. Your Starship (yellow arrow) is moved using the BBC micro's four cursor keys. The normal screen, or short range scan, gives one sector of an eight by eight area of space. Your sector position is given under 'AREA' (see short range scan, SRS, above left).

Two computers are under your control – nav-com and arm-con. To enter the first, press N on the keyboard, and hold it down for a couple of seconds. The screen will change to give information on energy and sector and four command choices:

- *Return to SRS;* this gives you back the normal screen.
- *See Cumulative Galaxy Record;* essentially says where you have been (above right). The three digits give first the number of Klingons, the number of stars (red * in SRS), and the number of Starbases in each sector you have explored.
- *Starbase data;* provides co-ordinates of bases found and energy left. Each contains up to 20,000 units of fuel. To dock at a Starbase (blue @) just position the Starship on top. Energy may be left for later.
- *Enter armaments computer.* This can be achieved also by pressing A when in short range scan.

The arm-com gives details on energy, photon torpedos fired (you have seven), shield level and number of Klingons destroyed. Its three commands are:

- *Return to SRS*
- *Fire photon torpedos;* these never miss when a Klingon is in range.
- *Reset shield; input energy up to 2000 units*
- *Enter nav-com.*

The shields give protection from Klingons, who automatically attack when you enter their sector. If you have no shields, your computers are put out of action for 30 seconds. Enemy hits are inversely proportional to shield strength, but each weakens the shields and costs energy. So the sooner you destroy the attacking Klingon, the better.

As well as the photon torpedos, you have phasers. But these must be aimed and fired in one of four directions by the keys 1(up), 2(right), 3(down), 4(left), when using the normal short range scan display.

When you meet a Klingon, or start running low on energy the screen will flash and give a STATUS RED report. If you try to leave the eight by eight grid, you find you can't and lose all shield energy.

Each torpedo and phaser uses 1000 units of energy. Each move uses ENERGY/1000 units, resetting shields uses SHIELD/40 units. Energy is the deciding factor, so use it wisely.

To play, LOAD "TREK III", wait for the cursor prompt > and then RUN.

Good luck, and happy hunting.



```
10 MODE7:*FX4,1
20 YC$=CHR$131:RC$=CHR$129:GC$=CHR$130
30 PRINTTAB(6,1)CHR$141;GC$;CHR$136"        T=R=E=K III":PRINTTAB(6,2)CHR$141;GC$;CHR$136" T=R=E=K III"
40 PRINTTAB(5,12)YC$"40 IS EASIER":PRINTTAB(2,10)YC$"INPUT DIFFICULTY (5-40) ";:INPUTD
50 VDU23,0,11,0,0,0:PRINTTAB(0,24)
60 ENVELOPE 1,3,2,2,2,50,30,30,126,-1,-1,-1,90,80:ENVELOPE 2,1,100,100,100,100,100,100,126,-50,-50,-50,120,90
70 DIM quadrant%(8,8),klingon%(15,2),base%(5,2),recharge%(5)
80 OK$=YC$+"ALL SYSTEMS OK       ":Dnav$=RC$+"NAV-COM DAMAGED      ":Darm$=RC$+"ARM-COM DAMAGED      ":NK$=YC$+"NO KLINGONS IN RANGE"
90 HK$=YC$+"KLINGON DESTROYED   ":LS$=RC$+"SHIELD LOW           ":LE$=RC$+"ENERGY LOW           ":R$=RC$+CHR$136+"RED "
100 G$=GC$+"GREEN"
110 star$=CHR$133+"*":phaser$=GC$+CHR$255:base$=CHR$134+"@":klingon$=RC$+"^":enterprise$=YC$+"^":blank$=GC$+CHR$255+CHR$255+CHR$255
120 status$=G$:report$=OK$:Dnav=0:Darm=0:Dkli=0
130 energy=D*10000:kl=15:NQ=0:B=0:move=0:shield=1000:phaser=0:photon=7:hit=0:torpedoe=0
140 navT=9999999:armT=9999999:klT=9999999
150 PROCinst:TIME=0
160 PROCsrs:PROCscreen
170 PROCsrsfill:PROCmove:PROCenergy:PROCcommand:PROCklingon:PROCcheck
180 GOTO170
190 DEFPROCinst
200 FORX=1 TO 8:FORY=1 TO 8
210 quadrant%(X,Y)=110
220 NEXTY:NEXTX
230 QposX=RND(8):QposY=RND(8)
240 SposX=RND(22)+3:SposY=RND(10)+4
250 FORL=1 TO 100
260 R1=RND(8):R2=RND(8)
270 quadrant%(R1,R2)=quadrant%(R1,R2)+10
280 IF L>15 GOTO 340
290 RX=RND(8):RY=RND(8)
300 quadrant%(RX,RY)=quadrant%(RX,RY)+100
310 klingon%(L,1)=RND(22)+3:klingon%(L,2)=RND(9)+4
320 IF L>5 GOTO 340
330 base%(L,1)=RND(8):base%(L,2)=RND(8)
340 NEXTL:ENDPROC
350 DEFPROCsrs
360 CLS:SOUND3,-15,200,1
370 PRINTTAB(1,3)CHR$150;CHR$55;TAB(1,15)CHR$150;CHR$117;TAB(28,3)CHR$107;TAB(28,15)CHR$122
380 FORZ=3 TO 27:PRINTTAB(Z,3)CHR$96;TAB(Z,15)CHR$112;:NEXTZ
390 FORZ=4 TO 14:PRINTTAB(1,Z)CHR$150;CHR$53;TAB(27,Z)CHR$150;CHR$106:NEXTZ:PRINTCHR$159
400 PRINTTAB(30,2)CHR$132;CHR$157;YC$;"TIME   ";CHR$156
410 PRINTTAB(29,5)CHR$131;CHR$157;RC$;"ENERGY  ";CHR$156
420 PRINTTAB(29,8)CHR$133;CHR$157;YC$;"STATUS  ";CHR$156
430 PRINTTAB(30,11)CHR$134;CHR$157;CHR$132;"AREA  ";CHR$156
440 PRINTTAB(29,14)CHR$130;CHR$157;YC$;"SHIELD  ";CHR$156
450 PRINTTAB(1,17)CHR$129;CHR$157;YC$;"STATUS REPORT:  ";CHR$156
460 ENDPROC
470 DEFPROCnavcom
480 IF Dnav=1 LET report$=Dnav$:GOTO170
490 CLS:SOUND3,-15,100,2
500 H$="NAVIGATION COMPUTER":PROCheadings
510 PRINTTAB(0,6)CHR$129;CHR$157;YC$;" ENERGY:"
520 PRINTTAB(0,7)CHR$134;CHR$157;CHR$132;" QUADRANT:";QposX;",";QposY
530 PRINTTAB(2,14)YC$;"<1> _ Return to S.R.S."
540 PRINTTAB(2,15)YC$;"<2> _ See cumulative galaxy record"
550 PRINTTAB(2,16)YC$;"<3> _ Star_base data"
560 PRINTTAB(2,17)YC$"<4> _ Enter armaments computer"
570 A$=INKEY$(10):A=VAL(A$)
```



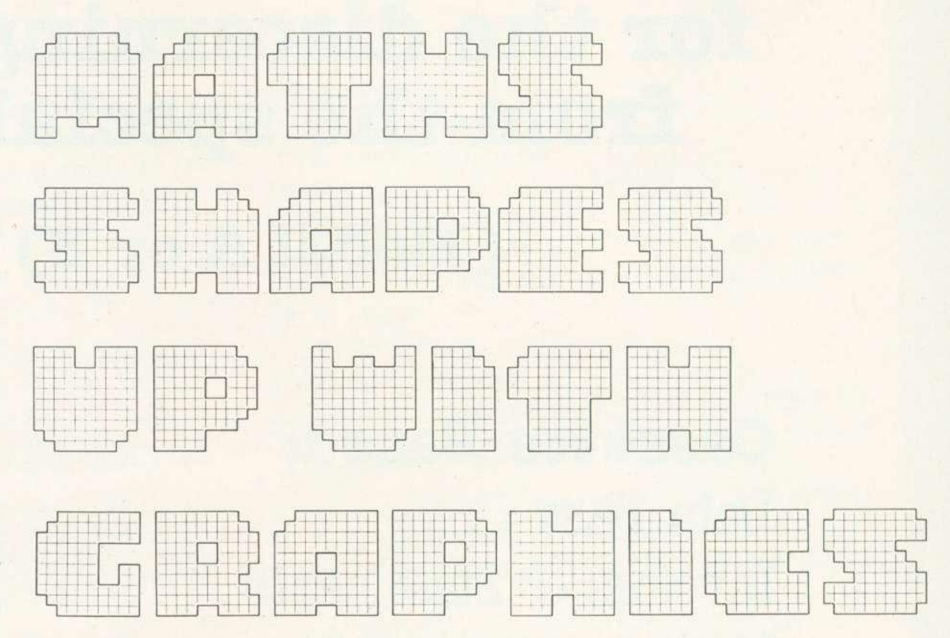
```
 58Ø *FX15,Ø
 59Ø PROCenergy:PROCklingon:PRINTTAB(1
5,6);energy
 6ØØ IF A<>1 AND A<>2 AND A<>3 AND A<>
4 GOTO57Ø
 61Ø ON A GOTO 16Ø,216Ø,24ØØ,64Ø
 62Ø ENDPROC
 63Ø DEFPROCarmcom
 64Ø IF Darm=1 LET report$=Darm$:GOTO1
7Ø
 65Ø CLS:SOUND 3,-15,15Ø,2
 66Ø H$="ARMAMENTS COMPUTER":PROCheadi
ngs
 67Ø PRINTTAB(Ø,6)CHR$129;CHR$157;YC$;
" ENERGY:"
 68Ø PRINTTAB(Ø,7)GC$;CHR$157;YC$;" PH
OTON TORPEDOES:"
 69Ø PRINTTAB(Ø,8)CHR$134;CHR$157;CHR$
132;" SHIELD:"
 7ØØ PRINTTAB(Ø,9)CHR$133;CHR$157;YC$;
" KLINGONS:"
 71Ø PRINTTAB(2,14)YC$;"<1> _ Return t
o S.R.S."
 72Ø PRINTTAB(2,15)YC$;"<2> _ Fire pho
ton torpedoes"
 73Ø PRINTTAB(2,16)YC$;"<3> _ Reset sh
ield"
 74Ø PRINTTAB(2,17)YC$;"<4> _ Enter na
vigation computer"
 75Ø B$=INKEY$(1Ø):B=VAL(B$)
 76Ø *FX15,Ø
 77Ø PROCenergy:PROCklingon:PROCarmfil
l
 78Ø IF B<>1 AND B<>2 AND B<>3 AND B<>
4 GOTO75Ø
 79Ø ON B GOTO 16Ø,8ØØ,82Ø,48Ø
 8ØØ IF LEFT$(STR$(quadrant%(QposX,Qpo
sY)),1)="1" report$=NK$:k1T=TIME:GOTO16
Ø
 81Ø photon=photon-1:torpedoe=1ØØØ:PRO
Chit:GOTO16Ø
 82Ø PROCshield:GOTO16Ø:ENDPROC
 83Ø DEFPROCsrsfill
 84Ø PRINTTAB(32,3)CHR$129;INT(TIME/1Ø
Ø)
 85Ø PRINTTAB(31,6)CHR$131;energy;" "
 86Ø PRINTTAB(3Ø,9)CHR$133;status$
 87Ø PRINTTAB(32,12)CHR$134;QposX;",";
QposY
 88Ø PRINTTAB(32,15)CHR$13Ø;shield;" "
 89Ø PRINTTAB(3,18)report$
 9ØØ ENDPROC
 91Ø DEFPROCenergy
 92Ø energy=INT(energy-(shield/4Ø)-pha
ser-torpedoe-hit-move):move=Ø:phaser=Ø:
hit=Ø:torpedoe=Ø
 93Ø ENDPROC
 94Ø DEFPROCcommand
 95Ø C$=INKEY$(Ø)
 96Ø *FX15,Ø
 97Ø IF C$="" GOTO1Ø3Ø
 98Ø SOUND3,-4,22Ø,1
 99Ø IF C$="1" OR C$="2" OR C$="3" OR
C$="4":SOUND3,-15,1ØØ,5:PROCphasers
 1ØØØ IF C$="N" PROCnavcom
 1Ø1Ø IF C$="A" PROCarmcom
 1Ø2Ø IF C$="S" PROCshield
 1Ø3Ø ENDPROC
 1Ø4Ø DEFPROCscreen
 1Ø5Ø IF MID$(STR$(quadrant%(QposX,Qpos
Y)),2,1)="1" GOTO1Ø9Ø
 1Ø6Ø FOR L=1 TO (VAL(MID$(STR$(quadran
t%(QposX,QposY)),2,1)))-1
 1Ø7Ø PRINTTAB(RND(22)+3,RND(1Ø)+4)star
$
 1Ø8Ø NEXTL
 1Ø9Ø IF LEFT$(STR$(quadrant%(QposX,Qpo
sY)),1)="1" GOTO 113Ø
 11ØØ FOR L=1 TO VAL(LEFT$(STR$(quadran
t%(QposX,QposY)),1))-1
 111Ø  PRINTTAB(klingon%(kl,1),klingon%
(kl,2))klingon$
 112Ø NEXTL
 113Ø ENDPROC
 114Ø DEFPROCmove
 115Ø NQ=Ø:PRINTTAB(SposX,SposY)"  "
 116Ø M=INKEY(Ø):IF M=-1 GOTO131Ø
 117Ø *FX15,Ø
 118Ø move=energy/1ØØØ
 119Ø IF M=136 SposX=SposX-1:enterprise
$="["
 12ØØ IF M=137 SposX=SposX+1:enterprise
$="]"
 121Ø IF M=138 SposY=SposY+1
 122Ø IF M=139 SposY=SposY-1:enterprise
$="^"
 123Ø IF SposX<3 QposX=QposX-1:SposX=25
:NQ=1
 124Ø IF SposX>25 QposX=QposX+1:SposX=3
:NQ=1
```

```
1250 IF SposY<4 QposY=QposY+1:SposY=14
:NQ=1
1260 IF SposY>14 QposY=QposY-1:SposY=4
:NQ=1
1270 IF QposX<1 QposX=QposX+1:SposX=3:
shield=0
1280 IF QposX>8 QposX=QposX-1:SposX=25
:shield=0
1290 IF QposY<1 QposY=QposY+1:SposY=14
:shield=0
1300 IF QposY>8 QposY=QposY-1:SposY=4:
shield=0
1310 PRINTTAB(SposX,SposY)enterprise$
1320 IF NQ=1 GOTO160
1330 ENDPROC
1340 DEFPROCklingon
1350 K=VAL(LEFT$(STR$(quadrant%(QposX,
QposY)),1))-1:IF K=0 GOTO 1440
1360 SOUND1,1,10,10:SOUND2,1,30,10:sta
tus$=R$
1370 FOR Z=1 TO K
1380 hit=RND(2300-shield):shield=INT(s
hield-(hit/23))
1390 IF RND(shield+5)=3 Dnav=1:navT=TI
ME
1400 IF RND(shield+5)=4 Darm=1:armT=TI
ME
1410 IF shield<0 Dnav=1:navT=TIME:Darm
=1:armT=TIME
1420 PRINTTAB(6,21)"          ":PRINTTAB(
2,21)YC$;"HIT:";hit
1430 NEXTZ
1440 ENDPROC
1450 DEFPROCphasers
1460 P=VAL(C$)
1470 IF P=1 X1=0:Y1=-1
1480 IF P=2 X1=1:Y1=0
1490 IF P=3 X1=0:Y1=1
1500 IF P=4 X1=-1:Y1=0
1510 phaser=1000:M2=0
1520 IF SposX+(M2*X1)=klingon%(kl,1) A
ND SposY+(M2*Y1)=klingon%(kl,2) PROChit
:GOTO1580
1530 PRINTTAB((SposX+((M2-1)*X1)),Spos
Y+((M2-1)*Y1));" ";TAB((SposX+(M2*X1))
,(SposY+(M2*Y1)));phaser$
1540 IF SposX+(M2*X1)>24 OR SposX+(M2*
X1)<4 OR SposY+(M2*Y1)>13 OR SposY+(M2*
Y1)<5 PRINTTAB(SposX+(M2*X1),SposY+(M2*
Y1))" ":GOTO1590
1550 M2=M2+1:SOUND3,1,200,1
1560 PROCenergy:PROCsrsfill:PROCklingo
n
1570 GOTO1520
1580 PRINTTAB(SposX+((M2-1)*X1),SposY+
((M2-1)*Y1))" "
1590 ENDPROC
1600 DEFPROCshield
1610 SOUND3,-15,180,1:MS=0:shield=0
1620 PRINTTAB(2,23)GC$;"SHIELD:"
1630 D$=INKEY$(0):S=VAL(D$)
1640 PROCenergy:PROCklingon
1650 IF B=3 PROCarmfill ELSE PROCsrsfi
ll
1660 IF D$="S" OR D$=""GOTO1630
1670 IF S<0 OR S>9 GOTO1630
1680 MS=MS+1:PRINTTAB(12,23)S
1690 IF MS=1 S=S*1000
1700 IF MS=2 S=S*100
1710 IF MS=3 S=S*10
1720 shield=shield+S
1730 IF MS=4 GOTO1750
1740 GOTO 1630
1750 IF shield>2000 shield=2000
1760 PRINTTAB(32,15)"      ";TAB(2,23
)"                "
1770 ENDPROC
1780 DEFPROChit
1790 SOUND2,2,100,10:SOUND0,-15,4,8:Dk
li=1:report$=HK$:klT=TIME:status$=G$
1800 quadrant%(QposX,QposY)=quadrant%(
QposX,QposY)-100
1810 PRINTTAB(klingon%(kl,1),klingon%(
kl,2))" "
1820 kl=kl-1:PRINTTAB(2,21)"
    "
1830 ENDPROC
1840 DEF PROCarmfill
1850 PRINTTAB(17,6)energy;TAB(25,7);ph
oton;TAB(17,8)shield;TAB(17,9)kl
1860 ENDPROC
1870 DEF PROCbase
1880 PRINTTAB(15,10)base$
1890 PROCenergy:PROCsrsfill:PROCmove
1900 IF SposX<>15 OR SposY<>10 GOTO196
0
1910 report$=YC$+"DOCKED:        "+STR$(
recharge%(L1))
1920 IF recharge%(L1)=20000 report$=RC
$+"DOCKED:NO FUEL     ":GOTO1880
```



```
1930 energy=energy+100:recharge%(L1)=r
echarge%(L1)+100
1940 SOUND1,-10,(20000-recharge%(L1))/
100,3:SOUND2,-10,(20000-recharge%(L1))/
101,3
1950 GOTO1880
1960 ENDPROC
1970 DEFPROCcheck
1980 IF kl=0 PROCwin
1990 IF energy<0 PROClose
2000 FORL1=1 TO 5
2010 IF QposX=base%(L1,1) AND QposY=ba
se%(L1,2) PROCbase
2020 NEXTL1
2030 IF RIGHT$(STR$(quadrant%(QposX,Qp
osY)),1)="0" quadrant%(QposX,QposY)=qua
drant%(QposX,QposY)+1
2040 report$=OK$:status$=G$
2050 IF VAL(LEFT$(STR$(quadrant%(QposX
,QposY)),1))>1 status$=R$
2060 IF shield<10 shield=0:report$=LS$
2070 IF Dkli=1 report$=HK$
2080 IF Dnav=1 report$=Dnav$
2090 IF Darm=1 report$=Darm$
2100 IF TIME>navT+3000 Dnav=0:navT=999
999:report$=OK$
2110 IF TIME>armT+3000 Darm=0:armT=999
999:report$=OK$
2120 IF TIME>klT+1000 Dkli=0:klT=99999
9:report$=OK$
2130 IF energy<20000 report$=LE$:SOUND
3,1,energy/100,5:status$=R$
2140 ENDPROC
2150 DEFPROCgalrec
2160 CLS
2170 H$="GALACTIC RECORD":PROCheadings
2180 FOR X=1 TO 8:FOR Y=1 TO 8
2190 SOUND1,-12,(X*30)+Y*2,1
2200 IF QposX=X AND QposY=Y col=135 EL
SE col=134
2210 R=VAL(RIGHT$(STR$(quadrant%(X,Y))
,1)):M=VAL(MID$(STR$(quadrant%(X,Y)),2,
1))-1:L=VAL(LEFT$(STR$(quadrant%(X,Y)),
1))-1
2220 Nbase=0
2230 IF R=0 PRINTTAB(X*4,(11-Y)*2)blan
k$:GOTO2260
2240 FOR L1=1 TO 5:IF X=base%(L1,1) AN
D Y=base%(L1,2) Nbase=Nbase+1:NEXT L1
2250 PRINTTAB((X*4),(11-Y)*2)CHR$col;L
;M;Nbase
2260 NEXT Y:NEXTX
2270 A=GET:GOTO160
2280 ENDPROC
2290 DEF PROClose
2300 CLS:PRINTTAB(2,3)"!!     YOU FOOL
 !!"'''"YOU RAN OUT OF FUEL IN ";TIME/
100;" secs"
2310 GOTO2370
2320 DEF PROCwin
2330 CLS:PRINTTAB(2,3)"!! WELL DONE CA
PTAIN !!"'''"YOU EXTERMINATED ALL THE K
LINGONS"
2340 score=INT((energy/(TIME/100))*(41
-D)):IF score>H% H%=score
2350 PRINTTAB(1,8)"STARBASE COMMAND AW
ARDS YOU ";score'" POINTS"
2360 PRINTTAB(5,12)"HI-SCORE=";H%
2370 PRINTTAB(2,20)"PRESS 'R' TO ERADI
CATE ANOTHER SECTOR"
2380 IF INKEY$(0)="R" RUN ELSE GOTO238
0
2390 DEF PROCbasedets
2400 CLS
2410 H$="STAR-BASE DATA":PROCheadings
2420 PRINTTAB(0,5)GC$+"STARBASE    POS
ITION   ENERGY AVAILABLE"
2430 FOR Z=1 TO 5:SOUND3,-10,Z*50,1:PR
INTTAB(1,(Z*2)+6)CHR$131;Z
2440 IF RIGHT$(STR$(quadrant%(base%(Z,
1),base%(Z,2))),1)="0" PRINTTAB(14,(Z*2
)+6);"     ":GOTO2460
2450 PRINTTAB(14,(Z*2)+6);base%(Z,1);"
,";base%(Z,2)
2460 PRINTTAB(30,(Z*2)+6);20000-rechar
ge%(Z)
2470 NEXTZ:A=GET:GOTO160
2480 DEFPROCheadings
2490 PRINTTAB(0,2)CHR$131;CHR$157;CHR$
132;CHR$141;TAB(6,2)H$
2500 PRINTTAB(0,3)CHR$131;CHR$157;CHR$
132;CHR$141;TAB(6,3)H$
2510 PRINTTAB(26,2)CHR$131;CHR$157;CHR
$129;CHR$136;CHR$140"ON-LINE";TAB(0,3)C
HR$131
2520 ENDPROC
```





Computer Graphics can put some fun into maths lessons for children in middle schools. Robin Norman explains how his class used a BBC machine

Open evening was looming and word went out that a contribution on 'shapes in maths' was required from each class. Then, at a staff meeting, came the great idea – let's get the children to design pictures made of triangles and program the smart BBC computer to draw them on the screen.

The decision taken, the design stage went without a hitch. We all laid out grids on squared paper, and drew X and Y scales going from 0 to 1250 and 0 to 900 respectively, in steps of 50. The Y axis was kept a little short to make room for names and titles. Programming would be easier, I suggested, if the corners of triangles were all on grid intersections; and we agreed that clever pictures would have some of their triangles joined edge to edge. And then we were off – stars, boats, birds, space shuttles and Snoopies . . . next time I will set a limit of five or seven triangles and make it a challenge!

While felt pens in their dozens were brightening up the pictures, I was considering how to bring the computer in. A BBC model A in mode 5 will paint pretty triangles in red, yellow or white, but the coding to do this is complicated for 12-year olds.

```
L.
    1 REM TRIPIC2. (C) 1982 R.J.NORMAN
  100 MODE5
  110    VDU23;8202;0;0;0:PRINT
  200 REPEAT
  210 PROCDATA
  220 UNTIL FALSE
10000  DEFPROCTRIPIC:LOCALA$,X%,Y%,C%,S%
10010  READname$:IFname$="END"ENDPROC
10020 READtitle$:PRINTTAB(0,1);name$;:S
      %=20-LENtitle$:IFS%<LENname$+2PRINT;
10030    PRINTTAB(S%);title$
10040 READA$
10050  IFA$="NEXT"PROCPAUSE:CLS:GOTO10010
10060  IFA$="M"READX%,Y%:MOVEX%,Y%:GOTO10040
10070   IFA$="P"READX%,Y%:C%=C%+1
10080 IFC%>3THENC%=1
10090 GCOL0,C%:PLOT85,X%,Y%:GOTO10040
10200 DEFPROCDATA
10210  RESTORE10230
10220 PROCTRIPIC
10230 DATA ANTHEA,FREDDY FISH,M,250,400,
      M,450,300,P,450,550,P,950,400,M,1050,30
      0,P,1100,550,NEXT
10240 DATA SNOOPY,YACHT,M,400,100,M, 125,250,
      P,1100,100,P,1150,250,  M,600,250
      ,M,300,300,P,600,800,M,600,900,P,750,85
      0,M,600,800,M,600,250,P,1150,300,NEXT
10300 DATA END
10310. ENDPROC
10400 DEFPROCPAUSE
10410 *FX15,0
10420 P%=INKEY(500)
10430 ENDPROC
```



I did want the class to feel they had done something towards the programming, so we hit on a compromise. Each child had to convert his picture to a data line containing his name, the picture title and a combination of the following commands

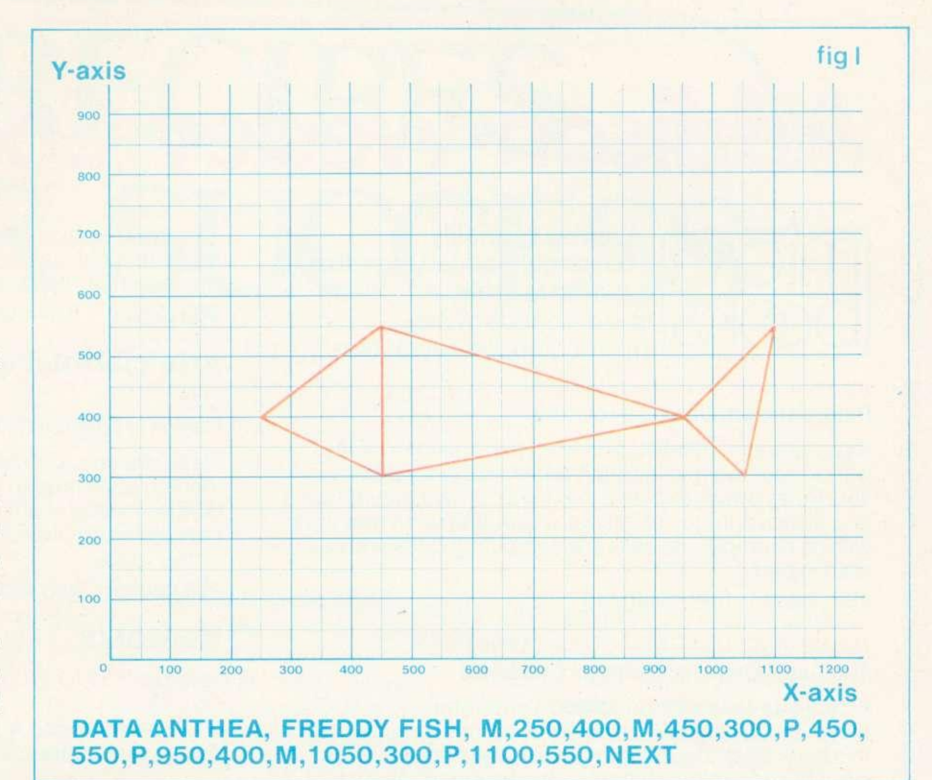
MOVE eg M,100,250

This means move to the screen position (100,250), and fix the apex of a triangle there.

PAINT eg P,500,100

This means move to the screen position (500,100), fix the apex of a triangle there, and paint the triangle formed by the last 3 positions given – whether these three were MOVE or PAINT did not matter.

NEXT means pause for five seconds to see the finished picture then clear the screen for the next one.



DATA ANTHEA, FREDDY FISH, M,250,400,M,450,300,P,450,550,P,950,400,M,1050,300,P,1100,550,NEXT

Figure 1 shows one of the simpler pictures with its DATA line. Each child worked out his or her own data, with help when needed, and we had half a dozen volunteers who typed them in to save time. It's very important to leave no trailing spaces after string DATA items – it is easy to do, especially after the final NEXT, and it spoils the pictures. Leading spaces are ignored, and so are those after numbers.

A backing program was obviously needed to convert the DATA lines into the instructions to make the BBC do what we wanted and it is shown here. You might like to add a title frame but don't use up too much of your precious RAM, which is in short supply in mode 5 on a model A.

Integer variables have been used throughout to save RAM – the program has room for about a dozen pictures, but you can save it on tape as often as you like with different sets of pictures. If you get bored with the colours, you can change them at will (or at random) by using the VDU 19 statement.

It was an enjoyable exercise and we got a lot from it by using the simple triangle – to make patterns and pictures.

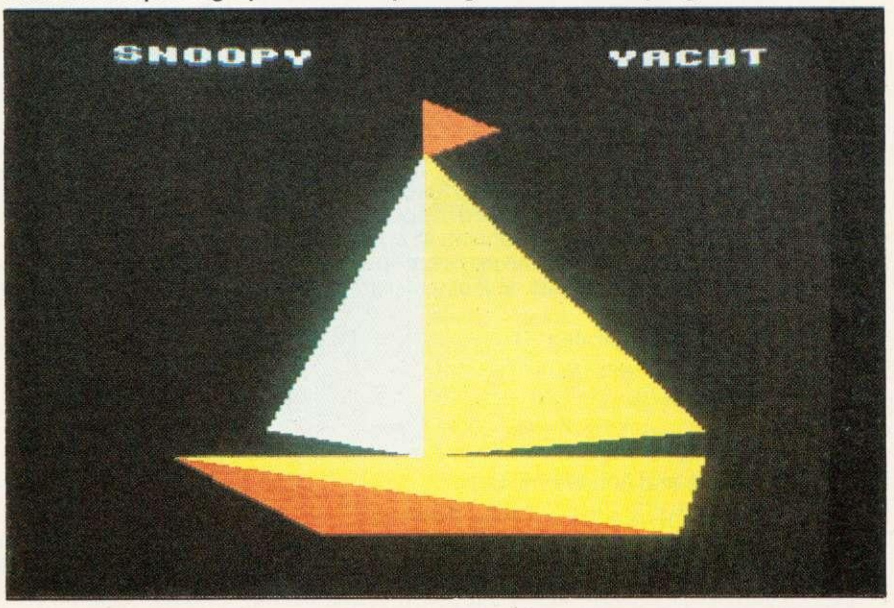
We also had a lot of practice in working out X and Y co-ordinates. The computer made it very obvious – by drawing silly pictures – where we had got them wrong.

And seeing how a computer can use a series of pairs of co-ordinates to draw a picture on the screen (the basis of all TV pictures) was extremely satisfying. Typing and editing the DATA lines was good practice and I had a ready-made PROC for drawing 'real pictures' (as opposed to random or mathematical patterns). I shall extend it to include a DRAW instruction for drawing lines and a bit more colour control.

It's not often that a maths teacher can honestly claim his pupils are enjoying their lessons, but our 'triangle project' for Open evening ensured fun and education for kids and staff alike – even if only for its duration.

And when the parents saw the masses of pretty pictures adorning the classroom wall they realised that we were actually doing something educational and different with the computer we had bought with their hard-earned cash!

This screen photograph was set up using data from the program above



RECIPES FOR SUCCESS

A computer without software is no good in a classroom. But who can write the programs? Jane Whitwell solved the problem by sitting two mothers in front of a machine with a set of recipes, or procedures to use as building blocks

Our first Acorn Atom arrived on six -month loan last January, but of the eleven members of staff and more than 300 excited children, only I had any previous experience of computers. Yet the major problem was yet to come. The new machine had no accompanying software!

By July we had trained two mothers of children in the school to be part-time programmers and written more than 20 programs

'Presentation on the screen is a major consideration'

suitable for our full primary age range, covering both language and mathematics.

We also had a computer club running every lunchtime and could boast proudly that almost every child in the school had used the machine – including the top two classes of our partner infant school.

This success was achieved by using a system we called 'recipe programming'. Here I produce short sections of code (usually in the form of subroutines) which perform a specific task – read in a child's name, draw a square, set up a two-dimensional array holding the days of the week, accompanied by a line by line 'explanation' and enough information, in the form of tables and charts to make alterations.

Our experience with children between the ages of six and twelve has shown that screen presentation is a major consideration in any program – particularly when dealing with the younger child whose reading ability may be restricted and whose memory span may be small.

We have been able to draw up a list of recommendations when using text on a TV screen:

- keep the text on the screen to a minimum.
- use short, concise messages, questions and answers.
- use plenty of spaces and delete old or superfluous text.
- do no write text too close to the screen edges as distortion effects can make it hard to read – especially when several children are gathered round a 14-inch screen.

Having had to return our first machine we are now looking forward excitedly to our very own BBC microcomputer and are eager to use the text/graphics window facility of this machine.

I have produced, therefore, a recipe which will produce a text window and a graphics window on a BBC model A micro. The window positions and dimensions were chosen to be suitable for even the youngest child when using a 14-inch screen:

```
10 MODE 5
20 VDU 24,5;5;500;500;
30 VDU 28,10,20,19,5
40 COLOUR 129
50 COLOUR 3
60 GCOL 0,130
70 CLS:CLG
```

The program can be explained as follows:

Line 10: The model A provides two graphics modes – 5 or 4. If this line is changed to mode 4 the effect is interesting but useless in this context.

Line 20 sets up the graphics window in the bottom left of the screen. Refer to figure 1 for a full account of the dimensions.

Line 30 sets up the text window in the top right of the screen.

'Keep text to a minimum and avoid edge distortion'

Refer to figure 1 for a full account of dimensions.

Line 40 sets up the background colour in the text window – in this case red. Refer to figure 2 to change colour.

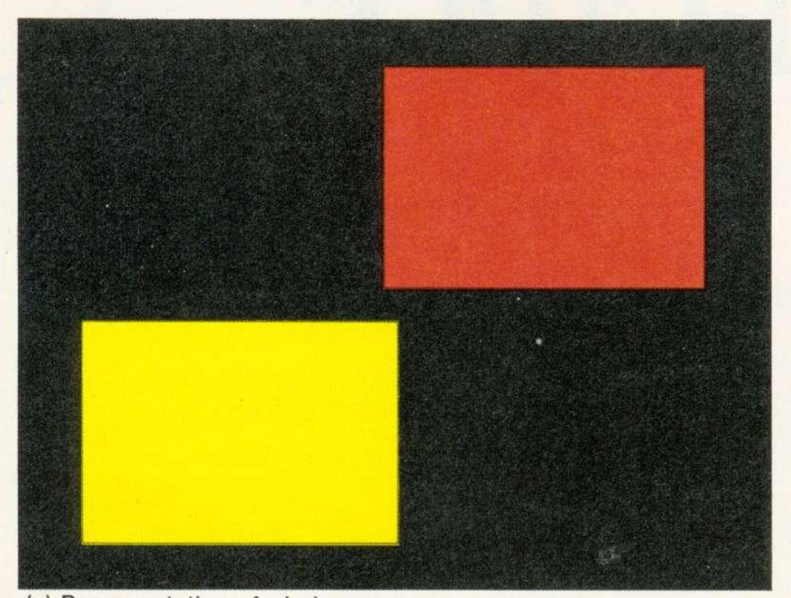
Line 50 sets up the foreground colour in the text window (the colour of the writing) in this case white. Refer to figure 2 to change colour.

Line 60 sets the background colour in the graphics window – in this case yellow. Refer to figure 2 to change colour.

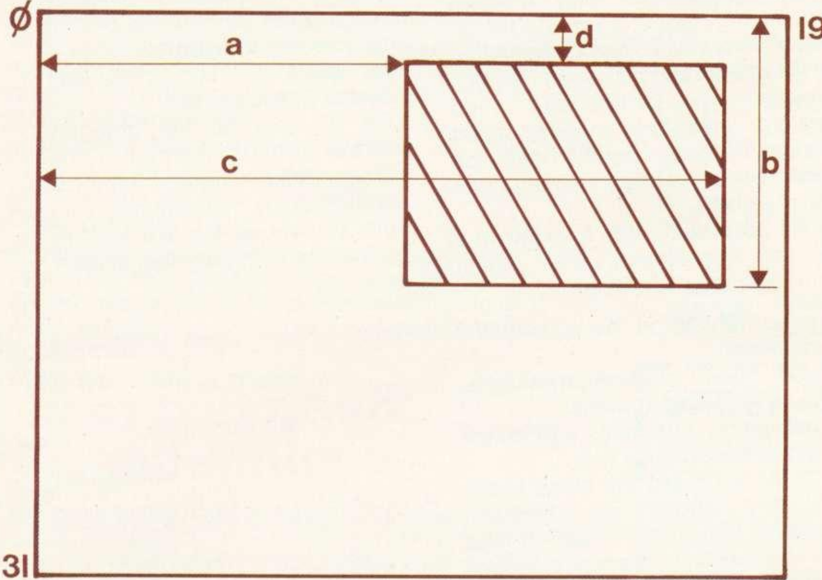
Line 70 clears the text window (CLS) and the graphics window (CLG).



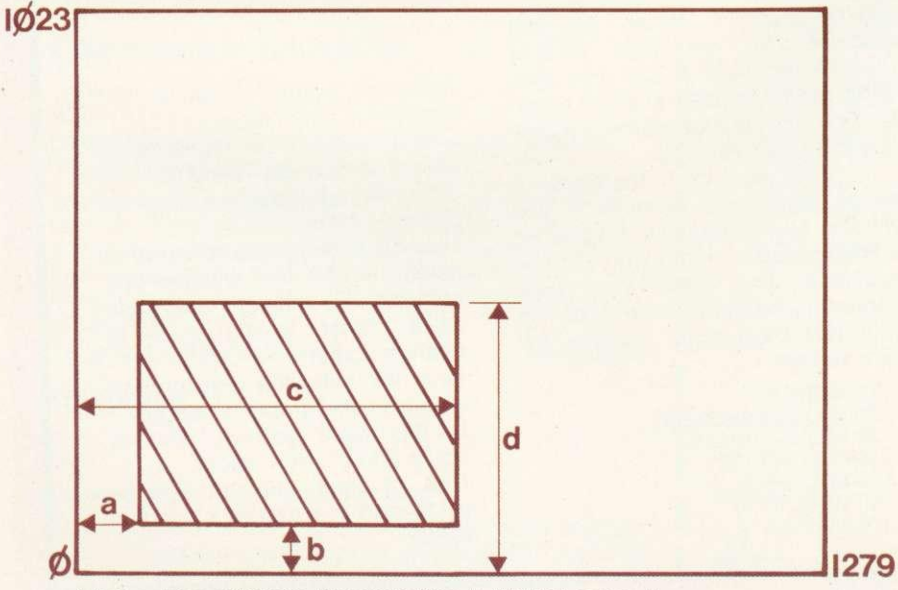
Figure 1. Relative size of windows in mode 5



(a) Representation of windows on screen



(b) Text window is (c − a + 1) characters wide by (b − d + 1) characters high. Program line is VDU28,a,b,c,d



(c) Graphics window. Program line is VDU24,a;b;c;d;

If typed in and run this recipe gives a red block and a yellow block on a black screen.

And if a one-colour printing loop is added to the window recipe:

```
 80 FOR N=1TO100
 90 PRINT "HELLO"
100 NEXT
```

the size and colour combinations can be seen in the text window. The white colour of the text (the foreground colour) was set by line 50 of the window receipe. To put up text in one colour and then another colour try:

```
 80 FOR N=1TO10
 90 PRINT "HELLO"
100 NEXT
110 COLOUR 2
120 FOR N=1TO10
130 PRINT "GOODBYE"
140 NEXT
```

and use figure 2 for changes to line 110.

Taking the window recipe (line 10 – line 70) a recipe can be added to draw a square within the graphics window:

'It is important that early experiences are successful'

```
180 MOVE 100,100
185 GCOL 0,0
190 DRAW 400,100
200 DRAW 400,400
210 DRAW 100,400
220 DRAW 100,100
```

Line 185 sets the colour of the lines of the square ie the 'graphics window foreground' colour – in this case black. Refer to figure 2 to change colour.

A useful recipe in this context is for the isosceles triangle which colours equal lines red and the base line black:

```
280 MOVE 100,100
285 GCOL 0,1
290 DRAW 200,400
300 DRAW 400,100
310 GCOL 0,0
320 DRAW 100,100
```

Figure 2 shows how to change line 285 and line 310 for other colour combinations.

	Text window		Graphics window	
	F'ground	B'ground	F'ground	B'ground
Black	COLOUR Ø	COLOUR 128	GCOL Ø,Ø	GCOL Ø,128
Red	COLOUR 1	COLOUR 129	GCOL Ø,1	GCOL Ø,129
Yellow	COLOUR 2	COLOUR 13Ø	GCOL Ø,2	GCOL Ø,13Ø
White	COLOUR 3	COLOUR 131	GCOL Ø,3	GCOL Ø,131

Figure 2. Recipe codes for various colour combinations

	Foreground (scores: 1 = poor contrast, 5 = good contrast)			
Background	Black	Red	Yellow	White
Black	invisible	3	5	5
Red	3 to 4	invisible	2 to 3	3 to 4
Yellow	4	2	invisible	1
White	4	2 to 3	1	invisible

Figure 3. Success of colour combinations

The square recipe can be used and then followed by the isosceles triangle with a CLG line inserted between the two. If this is omitted, the square is drawn first and the triangle is then inscribed.

Certain colour combinations give better results than others. If the window recipe (line 10 – line 70) is combined with the one-colour print loop (line 80 – line 100) and then the square recipe added (line 180 to line 220) a complete program can now be run to show all the effects. We have been able to draw up a combination table based on our results (figure 3).

Figure 4 gives the line numbers to be changed to produce different colour combinations and figure 2 gives the code.

We are sure the text/graphics window of the model A will have many uses in the programs we produce for the classroom. I am working on other recipes which, in conjunction with those outlined above, will form the basis of a series concerning geometric shape discrimination.

I am also devising, in association with an infant teacher, an exercise in simple co-ordinate plotting designed for a top infant class. And want to produce a series of programs concerned with the teaching of time – a complicated task which lends itself well to implementation on the BBC micro.

We find this method of producing software is quick and efficient – and this is vital for a busy teacher with little programming knowledge. It also makes my job easier when called on to debug another person's program.

Now, with the ever-increasing introduction of computers into schools, it is important the early experiences be successful. The teacher must add software designer and programmer to an already crowded job description. And any system which can make this easier can only increase computer use in the classroom. ●

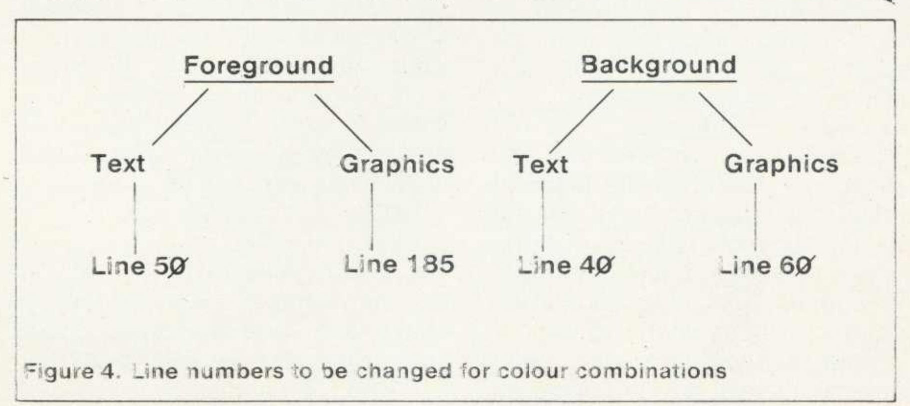


Figure 4. Line numbers to be changed for colour combinations



10 TEN PROGRAMS IN ONE MICRO

This isn't possible, you might think, but Michael Murray knows it can be done, and here he tells you how

The biggest handicap in mounting an effective microcomputer demonstration is the time taken to load programs from tape. For many, the simple solution – disc drives – is financially out of the question, so the only answer seems to be to have several programs in the machine at the same time.

The most obvious way to do this is to enter them as routines in one long program with a selection procedure, but this is both tedious and totally inflexible once the selection of programs to be combined has been decided. The other method involves entering a selection of different programs in separate sections of memory and jumping from one to another at will. The following procedures will enable this method to be carried out.

The basic principle is simple. After a program has been entered the page boundary is set above the top of it and the next program is then entered, and this may be repeated *ad infinitum* within the limits of the available memory.

However, in practice it is not quite so simple, so it is vital the procedures described are followed carefully or later programs will not function correctly. If an error is made it is likely that one or more of the programs will not function at all, or may run but be unable to list.

Once the programs have been correctly loaded it will appear to the user that the only program in the machine is the one being used, and in fact each program may be treated as if this is the case. Each program may be run, listed, or edited without affecting the others, as long as its length is not increased past the next page boundary, in which case it would corrupt the following program.

If NEW is used the machine will only forget the current program, and it may be restored by OLD as usual. The same applies to the BREAK, OLD procedure. However, if BREAK, OLD is used, the computer will return to the first program.

If any program is deleted or corrupted the others will not be affected and movement from one program to another can be achieved with a single function key.

Whether entering programs by keyboard or tape, it will be necessary between each program to determine TOP and set the new value of PAGE. So if many programs are to be entered it is easier to use a function key rather than entering similar instructions many times. This can be done by:

```
*KEY0"PAGE=TOP+&100|M
PRINT PAGE |M
```

The actual value of PAGE set will be the nearest page boundary above TOP. Page boundaries occur each Hex 100 ie &EOO, &FOO, &1000, &1100 and so on.

Before loading or entering the first program check that PAGE is at the default value of &EOO by entering:

```
PRINT~PAGE
```

If the answer is not EOO enter:

```
PAGE=&EOO
```

Load the first program in the usual way if you want to use tape. If the function key has been programmed correctly, press it and load the next program and repeat until all programs have been loaded. Each time the function key is pressed the new value of PAGE will be printed on the screen.

If the function key is not being used after each program has been loaded enter:

```
PRINT~TOP
```

Then determine and enter the new value of PAGE. For example, if TOP = 1E72 then the required new value will be PAGE = &1FOO. Whichever method is used, it is essential to keep a record of all the page boundaries.

To load from the keyboard, enter the first program in the usual way. Before loading a second program you must change the value of PAGE and then type NEW for the next program to be accepted.

This is most easily done be programming a function key as follows:

```
*KEY0"PAGE=TOP+&100 |M
PRINT PAGE |M NEW |M
```

Demonstration programs

```
*KEYØ"PAGE=TOP+&1ØØ|M
PRINT~PAGE |M NEW |M"

Program 1
10 PRINT"PROGRAM 1, PAGE=";~PAGE
20 PRINT"LOMEM=";~LOMEM

Program 2
Press fØ (note new PAGE=&FØØ)
10 PRINT"PROGRAM 2, PAGE=";~PAGE
20 PRINT"LOMEM=";~LOMEM

Program 3
Press fØ (note new PAGE=&1ØØØ)
10 PRINT"PROGRAM 3, PAGE=";~PAGE
20 PRINT"LOMEM=";~LOMEM
OLD
```

After each program has been entered it is then only necessary to press f0 (or whichever key was programmed) before entering the next. Keep a note of all PAGE values printed when the function key is pressed.

Because the computer normally sets LOMEM = TOP, if a program produces any variables (which are stored starting at LOMEM) it is possible for them to run over the next page boundary and corrupt the next program. This can be avoided by setting LOMEM = TOP of the last program entered (type OLD).

To go from one program to another all that is needed is to set the value of PAGE to the start of the program required, which is why it was necessary to keep a note of the PAGE boundaries. LOMEM was set when OLD was typed after the last program was entered and will stay there as long as the computer's pointers are not reset in any way. If the pointers are reset, LOMEM will be set to the top of which ever program is being accessed after the reset. This may be overcome by returning to the last program and resetting the pointers with the OLD command. As a safeguard this may be done before each program is run.

It is recommended that function keys are used for going from one program to another, one for each program loaded, as follows:

```
*KEYnumber"PAGE=Hex Address of
last program |M OLD |M
PAGE=Hex address of program
wanted |M RUN |M"
```

To show these procedures work, try the demonstration programs given.

To run the programs, use the function keys thus:

```
*KEY1"PAGE=&1000 |M OLD |M
PAGE=&E00 |M RUN |M"
*KEY2"PAGE=&1000 |M OLD |M
PAGE=&F00 |M RUN |M"
*KEY3"PAGE=&1000 |M OLD |M
RUN |M"
```

The underlined commands are not necessary in all circumstances, but are included in case the computer's pointers are ever reset by ESCAPE, OLD or NEW. In this case they are not necessary because they reset LOMEM above the last program and these programs create no variables, but they are included as an example.

Programs 1, 2 and 3 may now be run simply by pressing f1, f2 or f3. (It will be seen that LOMEM always stays above program 3 even if ESCAPE or BREAK, OLD are used – try it.) If the programs were such that they did not end, but ran continuously, then it would be necessary to press ESCAPE before the function key. Any number of programs may be entered as described as long as there is sufficient memory available. In practice it has been found that when running small demonstration programs the limiting factor is often not memory but the number of function keys.

Another, neater, method than the one above will work if there is space at the beginning of each program to insert an extra line, and none of the programs use the resident integer variable chosen (see below). It is basically the same as that already described except:

- After each program is entered, add an extra line at the beginning:
  1 LOMEM=L% (or any other resident integer variable).
- After the last program has been entered and OLD has been typed, type:
  L%=LOMEM

When the function keys are programmed they will now require less programming than before:

```
*KEYnumber"PAGE=Hex address of
program wanted |M RUN |M"
```

This is because each program itself will now set LOMEM above the last program, and the value they use, being held as resident integer variable, is protected from the effects of the commands that reset the computer's pointers.

Joe Telford sets up figures and landscapes for the BBC micro with the help of FOR NEXT loops to make your programs go . . .

FASTER AND FASTER

There are hardly any programs which have no use for a NEXT LOOP. Their value is that they allow a number of commands to be executed for a set number of times. When graphics in Basic seem slow because many parts of the screen are moving in different directions, FOR NEXT loops allow the programmer to speed up his programs.

```
10Z=TIME
20FOR X=1 TO 1000
30:
40REM THIS LOOPS AROUND 1000 TIMES
50:
60NEXTX
70PRINTTIME-Z;" /100 SEC"
```

Program 1. Points out crossed loops

Program 1 takes 1.28s to run (version 1.0 EPROMs) – which is slow for the BBC micro. Imagine taking this time to move a bullet up the screen! Although the colon lines and the REM line help to make the program understandable, they are executed 1000 times. Both a colon and a REM do nothing, but the computer has to interpret these lines.

To maintain readability the REMs should be placed outside the loop. Re-writing lines 30, 40, 50, will speed up the loop to take 0.64 seconds – twice as fast. This speed gain can be further increased, although the next step does reduce readability.

When the computer executes a FOR NEXT loop, it translates the NEXT line and looks for the variable at the end of the line – X, in Program 1. The program then checks that this variable matches the one in the corresponding FOR line. If it does match, the loop continues; if not, an error message is generated. This is because while nested loops are allowed:

```
 10 FOR X = 1 To 10
 20 FOR Y = 1 TO 5
120 NEXT Y
130 NEXT X
```

crossed loops are not:

```
10 FOR X = 1 TO 10
20 FOR Y = 1 TO 5
30 NEXT X
40 NEXT Y
```



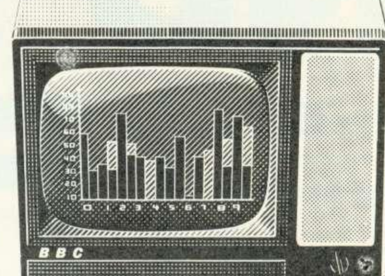
But, if we lose the variable from the NEXT lines, the computer automatically assumes that each NEXT line belongs to the last incompleted FOR line. This clears the way for a further speed increase because no check needs to be made.

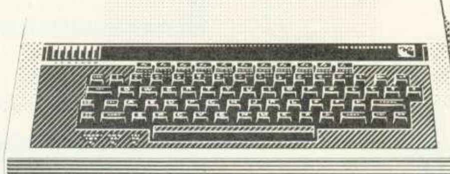
Altering line 60 to read 60 NEXT reduces the execution time to 0.5s. The final stage is to alter the control variable in line 20 to 20 FOR X% = 1 TO 1000. X% is an integer variable, and integers – whole numbers – are stored in a representation which allows addition to be performed speedily. Integer loops, though, cannot handle

MICROAGE ELECTRONICS

BBC Microcomputers
'B's and expanded 'A'
IN STOCK NOW!

BBC Model 'B' wordprocessing pack at a low price of **only £699. Save £44.** Normal price £743. The Pack consists of: BBC Model 'B' GP100 Printer Cables, Cassette Player Word Processing ROM 1,000 sheets of paper. Then add the GP100A Printer at only £215. The lowest price ever.





BBC MACHINES

Model A, 32K RAM + 6522 Chip	£329.00
Model B	£399.00
Model B + Disk Interface	£494.00
BBC Dust Covers	£3.95 + £1.00 p&p
BBC Compatible Single Disk Drive (100K)	£235.00
BBC Compatible Dual Disk Drive (200K)	£389.00
BBC Dual Slimline Disk Drive (Double Sided & Density 800K)	£799.00
Verbatim Single Sided Diskettes 10 for	£22.50
Verbatim Double Sided Diskettes 10 for	£39.99
Let us fit a disk interface in 24hrs	£95.00

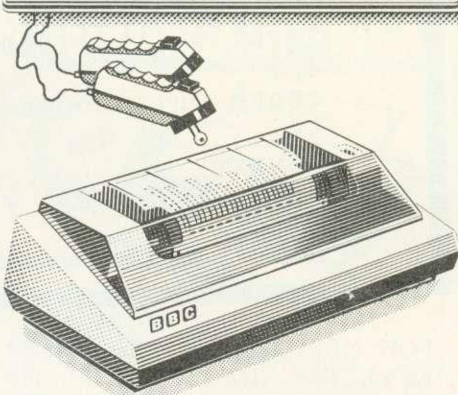
BBC MONITORS

14" RGB Microvitec Colour Monitor (as used in the BBC Computer Prog.)	£284.00
RGB Lead for above	£4.95
12" Zenith High Res. Green screen Monitor	£95.00
BNC Cable for above	£4.95

BBC CASSETTE PLAYER

Official Version inc. Leads, tape counter, Battery/Mains etc.	£28.00 + £2.00 p&p
Blank Data Cassettes 10 for	£3.50 + £1.30 p&p
DIN to Jack Lead	£3.50 + £1.00 p&p
Official Joysticks per pair	£13.00 + £1.30 p&p

Machines, Disks, Monitors and Printers Delivered by Courier 1 item **£7**, 2 items **£10**, 3 or more **£13**



PRINTERS

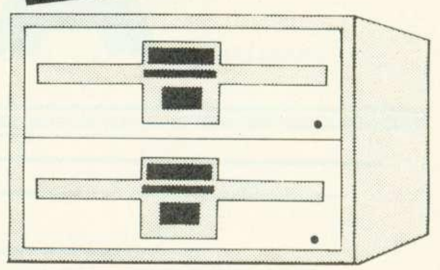
Acorn AP-80A now down to	£189.00
Acorn AP-100A now down to	£215.00
AP Ribbons	£4.95 + £1.00 p&p
Epson MX-80 F/T 111 (new model, Dot matrix High res. graphics, 80 or 132 chars. per line	£390.00
Prices include cable & paper	
Epson Dust Cover	£4.95 + £1.00 p&p
Parallel printer cable	£15.00

ACORNSOFT FOR BBC

Snapper, Planetoid, Monsters, Rocket Raid, Meteors, Philosophers Quest, Sphinx Adventure, Arcadians, Chess, Lisp Cassette, Forth Cassette, Business Games, Peeko-Computer, Creative Graphics Tape, Graphs & Charts Tape, Desk Diary, Arcade Action, View (on ROM)

All Acornsoft at £9.95 each, except Arcade Action (11.90), Forth & Lisp (16.85 each) and View (£59.95).

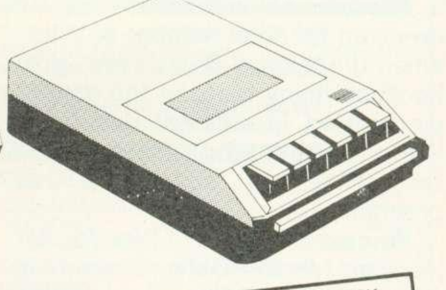
THE ACORN ATOM
Large stocks of all Atom kits, machines, disk-packs, ROMs and accessories. Software includes Attomcalc, FORTH, LISP, Adventure etc. Ring or write for list, prices etc.



BOOKS

Practical Programs for BBC & Atom	**£5.95**
BASIC Programming on the BBC Micro	£5.95
Programming the 6502 (Zaks)	£10.75
6502 Assembly Language Programming	£11.95
BBC Micro Revealed	£7.95
Creative Graphics, Graphs & Charts, LISP & FORTH	all at **£7.50** each

Postage & packing for software & books
1 item **£1.00**
2 items or more add **50p** per unit



SPECIAL WORD PROCESSING PACK – SAVE £150!!
BBC 'B' with 800k disk drive, word processing ROM and Epson printer – list price £1,752. **Our price £1,599.**



Barclaycard and Access welcomed All prices include VAT

We accept official orders from educational establishments. We specialise in mail-order, all over the world – send large S.A.E. for lists and info pack.

24hr answerphone for credit-card holders. Phone in for 'express' despatch.

ALL PRICES INCLUDE VAT. FOR FURTHER DETAILS AND MAIL ORDER LIST SEND LARGE S.A.E.
Open Mon - Sat 9am - 6pm. Thurs 9am - 1pm

MICROAGE ELECTRONICS
135 HALE LANE EDGWARE MIDDLESEX HA8 9QP
TEL: 01-959 7119 TELEX 881 3241

```
5DIMTT$(50,8,5)
10Z=TIME
20FOR T=1 TO 50
25REM FOR 50 TEACHERS
30FOR D=1 TO 8
35REM FOR 5 DAYS
40FOR P=1 TO 5
45REM FOR 8 PERIODS
50REM INITIALISE EACH TT LOCATION
80TT$(T,D,P)="0000"
100NEXT P
110NEXT D
120NEXT T
130PRINTTIME-Z;" /100 SEC"
```

```
5DIMTT$(50,8,5)
10Z=TIME
20FOR T%=1 TO 50
30FOR D%=1 TO 8
40FOR P%=1 TO 5
80TT$(T%,D%,P%)="0000"
100NEXT:NEXT:NEXT
130PRINTTIME-Z;" /100 SEC"
```

Program 2. Simulates teaching timetable

Program 3. Same function, but faster

decimals. For example:

FOR X% = 1 TO 10 STEP.1

will not work.

The loop is now reduced to 0.19 seconds – about one-seventh of the original loop speed. Programs 2 and 3, which simulate initialising a timetable for 50 teachers over a week of eight lessons per day, put this saving into perspective. Program 2 executes in 9.5 seconds.

Applying the techniques suggested so far gives us Program 3 – which runs in half the time, and has a larger memory.

The BBC computer can load spaces in program lines.

Type 20 _ _ _ _ _ _

The _'s are single spaces. List the program and you will notice line 20 appears. Spaces use up memory and waste time. To look for spaces in a long program simply load the program, enter mode 4, and type

VDU23,32,255,255,255,255,255,255,255,255

Now list the program. All spaces will be shown in white, and it is up to you whether or not to remove them. White spaces before line numbers cannot be removed.

As I illustrated last month, printing and moving shapes across the screen, is easy, provided characters are rationed to one square per shape. But there are many applications where shapes need to be built around more than one character.

Two techniques can be used, which both rely on defining characters that can be printed together to form the compound shape. Once these characters have been defined the first method prints them sequentially line by line. The second works in the same way as for single characters – printing the whole shape at once. This has the edge over the first method because it is quicker and affords control of printing position.

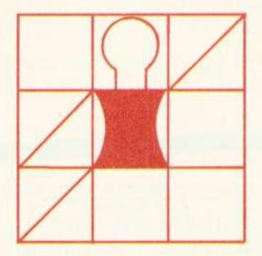
Figure 1 shows the compound shape for a figure made up of nine squares incorporating five different characters. These characters, must first be defined in VDU 23 statements, with the exception of the space character which already exists.

Figure 2 illustrates how the figure is redrawn before being coded into a string. The gaps and extensions to the right are for control characters which organise the layout of a string into the two-dimensional picture of figure 1.

Figure 3 is the final representation for moving the figure left or right. It has the extra spaces at the edges to erase the character in whichever direction it moves. Note that because spaces are not added above and below, the shape will leave an assorted scattering of links if moved up or down.

Program 4 creates a little figure which scampers across the screen at a touch of the > and < keys. Lines 100 to 150 define the separate squares. Line 160 contains the control codes to print the shape as a two-dimensional figure.

MAN1$ and MAN2$ are two version of the figure. Once the


Figure 1. Compound shape

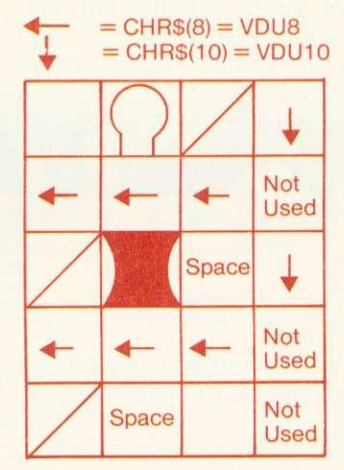


Figure 2. Shape redrawn for string

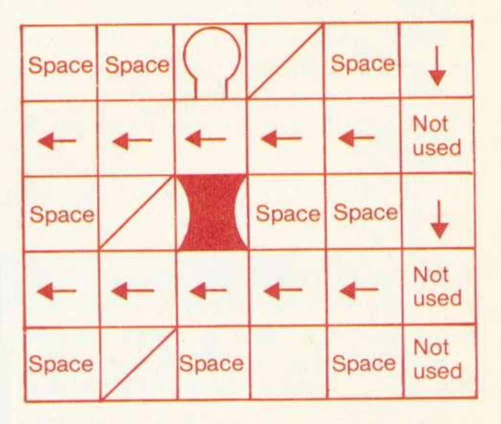


Figure 3. Final version for moving shape



Program 4. Scampering character

```
  1ØØVDU23,224,1,2,4,8,16,32,64,128
  11ØVDU23,225,128,64,32,16,8,4,2,1
  12ØVDU23,226,24,1Ø2,129,129,129,1Ø2,2
4,24
  13ØVDU23,227,255,255,126,6Ø,6Ø,126,25
5,255
  14ØVDU23,228,1,1,1,1,1,1,1,1
  15ØVDU23,229,128,128,128,128,128,128,
128,128
  16ØDB$=CHR$(1Ø)+STRING$(5,CHR$(8))
  17ØMAN1$="  "+CHR$(226)+CHR$(224)+" "
  18ØMAN1$=MAN1$+DB$+" "+CHR$(224)
  19ØMAN1$=MAN1$+CHR$(227)+"  "
  2ØØMAN1$=MAN1$+DB$+" "+CHR$(224)
  21ØMAN1$=MAN1$+" "+CHR$(229)+" "
  23ØMAN2$=" "+CHR$(225)+CHR$(226)+"  "
  24ØMAN2$=MAN2$+DB$+"  "+CHR$(227)
  25ØMAN2$=MAN2$+CHR$(225)+" "
  26ØMAN2$=MAN2$+DB$+" "+CHR$(228)+" "
  27ØMAN2$=MAN2$+CHR$(225)+" "
  29ØMODE4:CLS
  3ØØX=1:Y=16:PROC_FWD
  31ØA$=GET$
  32ØIFA$=">" OR A$="."THEN PROC_FWD
  33ØIFA$="<" OR A$=","THEN PROC_BWD
  34ØGOTO31Ø
  35ØDEFPROC_FWD
  36ØIF X+1=34THEN ENDPROC
  37ØPRINTTAB(X,Y);MAN2$
  38ØPROC_DELAY
  39ØX=X+1
  4ØØPRINTTAB(X,Y);MAN1$
  41ØPROC_DELAY
  42ØENDPROC
  43ØDEFPROC_BWD
  44ØIF X-1=ØTHEN ENDPROC
  45ØPRINTTAB(X,Y);MAN1$
  46ØPROC_DELAY
  47ØX=X-1
  48ØPRINTTAB(X,Y);MAN2$
  49ØPROC_DELAY
  5ØØENDPROC
  51ØDEFPROC_DELAY
  52ØLOCAL Z
  53ØZ=TIME+1Ø
  54ØREPEAT UNTIL TIME>Z
  55ØENDPROC
>
```

program is run, use > or < to move the figure. This is done via the two procedures 'FWD' and 'BWD' at lines 350 and 430 respectively.

To print such a figure on the screen, simply PRINT TAB the string containing the shape to the new position.

Movement must be one square at a time,

```
PRINT TAB(10,15);MAN1$
PRINT TAB(11,15);MAN2$
```

is fine but

```
PRINT TAB(10,15);MAN1$
PRINT TAB(12,15);MAN2$
```

is not. Aan assortment of links would be scattered in the wake of the figure.

Program 5 is a more efficient routine for creating the figures. It simply replaces the lines 160 to 270 in Program 4.

My preference is for the first version, although as I become better at saving space I shall probably move to the Program 5 routine.

Program 6 in its subsections is a complete game using sound, colour and moving graphics. The procedures are REMed, and so require little explanation. The aim of the game is to destroy the yellow building placed at random on the surface. Simply press the space key to drop the one bomb allotted to each flight.

Program 7 draws three circles in flashing colours, so that the finished pattern pulsates. It must be run in mode 2. Program 8 converts Program 7 to be used as the background to an advertising board. The name of the advertiser fits into A$ in line 370 and the products being sold follow on from line 820.

Flashing colours are the first step to animated graphic shapes. To achieve this effect, the shape required is plotted; using DRAW, MOVE, and PLOT 85 commands, but is coded by a VDU 19 command to be the same colour as the background.

The shape can then be plotted in several positions, and because the plotting is all in the background colour, the screen is left clear.

To move the shape simply use a VDU 19 command to alter the colour of the shapes in turn, making each stand out from the background, and making the last shape the same colour as the background.

The number of shapes depends on the number of colours available. In mode 2, there are eight non-flashing colours, so seven shapes could be plotted identically – although coded to be background colours.

The shapes would then flash into existence immediately on execution of a VDU 19 command. One of the most useful areas for this facility is in the apparent expansion/contraction of a shape on the screen. ▶ page 40

Program 5. More efficient technique for program 4

```
17ØDATA32,32,226,224,32,1Ø,8,8,8,8,8
18ØDATA32,224,227,32,32,1Ø,8,8,8,8,8
19ØDATA32,224,32,229,32
2ØØMAN1$="":FORX%=1TO27
21ØREAD J:MAN1$=MAN1$+CHR$(J)
22ØNEXT
23ØDATA32,225,226,32,32,1Ø,8,8,8,8,8
24ØDATA32,32,227,225,32,1Ø,8,8,8,8,8
25ØDATA32,228,32,225,32
26ØMAN2$="":FORX%=1TO27
27ØREAD J:MAN2$=MAN2$+CHR$(J)
28ØNEXT
>
```



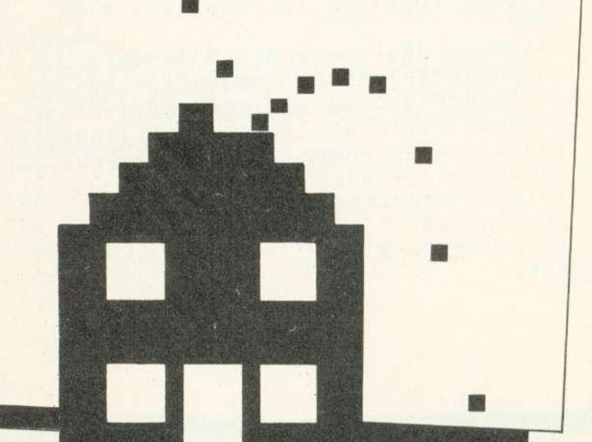
Program 6. The aim of the game is to destroy the yellow building using a bomb. It is well REM'd and explained in sections

```
100REM BOMBER
110REM J.TELFORD
120REM SIMPLE DEMO OF
130REM MOVING GRAPHICS
140REM IN A GAME FORMAT.
150:
160REM MAIN GAME
170:
180ON ERROR GOTO360
190PROC_TITLE
200PROC_INITIALISE
210REPEAT
220H=0
230REM** START GAME**
240MODE5
250FOR Plane = 1 TO NF
260PROC_Setupflight
270PROC_Drawland
280PROC_Flight_Info
290PROC_Early_Warning
300PROC_Move_Plane
310NEXT Plane
320MODE7
330PROC_RESULTS
340PROC_AGAIN
350UNTIL N%
360MODE7:PRINT''"Bye":END
370:
380:
390:
400DEFPROC_DROP
410:
420REM BOMB FALLS FROM PLANE
430REM TO GROUND.
440:
450FORBD=3TO24
460PRINTTAB(PP+1,BD);B$
470:
480REM WITH SOUND
490:
500PROC_WHISTLE
510:
520REMEMBER TO ERASE IMAGES OF BOMB
530:
540PRINTTAB(PP+1,BD);" "
550NEXT
560:
570REM SET UP EXPLOSION COLOUR
580REM FLASHING RED/CYAN
590:
600VDU19,1,9;0;:COLOUR1
610:
620REM AND PRINT EXPLOSION
630:
640PRINTTAB(PP,25);EX$
650:
660REM SET NORMAL COLOUR
670REM BEFORE I FORGET
680:
690COLOUR3
700:
710REM CHECK FOR A YELLOW TARGET
720REM IF IT HAS A DIFFERENT COLOUR
730REM THEN TARGET IS HIT.
740:
750IFPOINT(LONGIT,UPIT)<>3 THEN PROC_
WIN ELSE SOUND &10,1,107,30
760:
770REM ONLY 1 BOMB PER FLIGHT
780:
790BOMB=0
800ENDPROC
810DEFPROC_WHISTLE
820SOUND&0011,(-3+(BD/2*-1)),150-BD,1
830SOUND&0012,(-3+(BD/2*-1)),150-BD,1
840ENDPROC
850DEFPROC_INITIALISE
860:
870REM SET FAST FLASHING RATES
880:
890*FX10,2
900*FX9,2
910:
920REM SET COLOURS UP
930:
940VDU19,0,7;0;
950VDU19,1,9;0;
960VDU19,2,2;0;
970VDU19,3,7;0;
980VDU19,128,4;0;
990:
1000REM SET EXPLOSION ENVELOPE
1010:
1020ENVELOPE1,1,0,0,0,0,0,0,5,126,0,-5
,40,126
1030COLOUR128
1040:
1050REM SET 10 FLIGHTS
1060:
1070NF=10
1080:
1090REM DEFINE GRAPHICS FOR THE
1100REM 4 MAIN PROGRAM SHAPES.
1110:
1120PROC_PLANE
1130PROC_BOMB
1140PROC_LANDSCAPE
1150PROC_EXPLOSION
1160ENDPROC
1170:
1180DEFPROC_PLANE
1190:
1200REM DEFINE SINGLE CHR GAPHICS
1210REM TO MAKE UP PLANE
1220:
1230VDU23,224,192,224,240,248,255,129,
255,255
1240VDU23,225,0,0,0,0,255,240,255,255
1250VDU23,226,0,0,24,60,255,3,255,255
1260VDU23,227,0,0,0,0,192,240,240,192
1270:
1280REM PUT TOGETHER INTO A$
1290:
1300A$=" "+CHR$(224)+CHR$(225)
1310A$=A$+CHR$(226)+CHR$(227)
1320ENDPROC
1330:
```

```
1340DEFPROC_BOMB
1350:
1360REM DEFINE BOMB SHAPE
1370:
1380VDU23,228,65,42,28,8,28,62,28,8
1390:
1400REM PUT INTO B$ FOR EASY USE
1410:
1420B$=CHR$(228)
1430ENDPROC
1440:
1450DEFPROC_LANDSCAPE
1460:
1470REM DEFINE SINGLE CHR GRAPHICS
1480REM TO MAKE UP LANDSCAPE
1490:
1500VDU23,229,0,0,0,0,255,255,255,255
1510VDU23,230,0,0,0,0,0,0,0,0
1520VDU23,231,255,255,255,255,255,255,
255,255
1530VDU23,232,8,28,42,8,28,42,8,8
1540VDU23,233,192,192,192,192,255,255,
255,255
1550VDU23,233,192,192,192,192,255,255,
255,255
1560VDU23,234,7,7,7,7,255,255,255,255
1570:
1580REM MAKE THEM AVAILABLE IN AN
1590REM ARRAY CALLED LS$( )
1600:
1610DIMLS$(6):FORX=1TO6:LS$(X)=CHR$(22
8+X):NEXT
1620ENDPROC
1630:
1640DEFPROC_EXPLOSION
1650:
1660REM DEFINE SINGLE CHR GRAPHICS
1670REM TO MAKE UP EXPLOSION
1680:
1690VDU23,235,255,255,255,0,0,0,0,0
1700VDU23,236,4,2,41,24,70,49,12,255
1710VDU23,237,28,28,28,62,62,190,127,2
55
1720VDU23,239,28,28,28,28,28,28,28,28
1730VDU23,240,28,127,127,255,255,127,6
2,28
1740VDU23,241,28,126,255,255,255,255,1
26,60
1750VDU23,242,28,254,254,255,255,254,6
2,28
1760VDU23,238,16,32,74,140,48,198,24,2
55
1770:
1780REM PUT SINGLE CHR GRAPHICS
1790REM TOGETHER TO FORM EXPLOSION
1800:
1810EX$=CHR$(236)+CHR$(237)+CHR$(238)
1820EX$=EX$+CHR$(11)+CHR$(8)+CHR$(8)
1830EX$=EX$+CHR$(239)+CHR$(11)+CHR$(8)
1840EX$=EX$+CHR$(239)+CHR$(11)+CHR$(8)
1850EX$=EX$+CHR$(239)+CHR$(11)+CHR$(8)
1860EX$=EX$+CHR$(8)+CHR$(240)
1870EX$=EX$+CHR$(241)+CHR$(242)
1880ENDPROC
1890DEFPROC_TITLE
1900:
1910REM YELLOW, DOUBLE HEIGHT
1920:
1930CLS
1940PRINT TAB(14,10);CHR$141;CHR$131;"
Bomber"
1950PRINT TAB(14,11);CHR$141;CHR$131;"
Bomber"
1960PROC_SPACE
1970ENDPROC
1980DEFPROC_Setupflight
1990BOMB=-1:REM THERE IS A BOMB
2000:
2010REM BLUE SKY
2020:
2030VDU19,128,4;0;
2040COLOUR128:CLS
2050:
2060REM LAND WILL BE GREEN
2070:
2080VDU19,2,2;0;
2090COLOUR2
2100ENDPROC
2110DEFPROC_Drawland
2120:
2130REM RANDOM SHAPES IN GREEN
2140REM DRAWN ACROSS SCREEN
2150:
2160FORX=0TO19
2170PRINTTAB(X,25);LS$(RND(6))
2180PRINTTAB(X,26);CHR$(235)
2190NEXT
2200:
2210REM RANDOM POS. FOR TARGET
2220:
2230PT=RND(14)
2240:
2250REM TARGET IS YELLOW
2260:
2270VDU19,3,3;0;:COLOUR3
2280:
2290REM PRINT TARGET
2300:
2310PRINTTAB(PT+3,25);CHR$(231)
2320:
2330REM FIND POS. OF TARGET
2340REM IN GRAPHICS NUMBERS
2350:
2360LONGIT=1280/20*(PT+3)+32
2370UPIT=1024/32*(32-25)-16
2380ENDPROC
2390:
2400DEFPROC_Flight_Info
2410PRINTTAB(1,28);"FLIGHT NO: ";Plane
2420PRINTTAB(1,30);"HITS:-          ";H
2430ENDPROC
2440:
2450DEFPROC_Early_Warning
2460FORWT=1 TO 1000:NEXT
2470VDU7:REM READY TO FLY
2480FORWT=1 TO 2000:NEXT
2490ENDPROC
2500:
2510DEFPROC_Move_Plane
2520:
2530REM PLANE IS YELLOW
2540:
```

▶ page 42

```
2550COLOUR3
2560:
2570REM ANTICIPATION NOT ALLOWED
2580:
2590*FX15,0
2600:
2610REM MOVE PLANE POSITION (PP)
2620REM ACROSS SCREEN
2630:
2640FORPP=1TO15
2650PRINTTAB(PP,2);A$
2660:
2670REM GET A KEY
2680:
2690BD$=INKEY$(0)
2700:
2710REM TEST FOR DROP BOMB COMMAND
2720REM AND FOR A BOMB TO DROP
2730:
2740IFBD$=" "AND BOMB THEN PROC_DROP:G
OTO2800
2750:
2760REM IF NOT DROPPED WAIT THEN
2770REM CONTINUE.
2780:
2790FOR WT=1TO100:NEXT
2800NEXT
2810:
2820REM BLANK LAST PLANE IMAGE
2830:
2840PRINTTAB(PP,2);"    "
2850:
2860REM RELAX AT END OF A FLIGHT
2870:
2880FOR WT = 1 TO 2000:NEXT
2890ENDPROC
2900 DEFPROC_SPACE
2910 PRINTTAB(3,24);"Press the SPACE B
AR to continue";:*FX15,0
2920 REPEAT UNTIL GET$=" "
2930 ENDPROC
2940:
2950DEFPROC_RESULTS
2960CLS
2970PRINT''''"You had ";H;" hits,";
2980PRINT" from ";NF;" flights."
2990IF H = NF THEN PRINT'" Excellent w
ork"
3000IF H = NF-1 OR H = NF-2 THENPRINT'
" Very good work"
3010IF H = NF-3  OR H=NF-4 THENPRINT'"
Good work"
3020IFH=NF-5 OR H=NF-6 THENPRINT'" Ave
rage!!!"
3030IFH=NF-7 OR H=NF-8 THENPRINT'" You
need practice"
3040IFH=NF-9 OR H=NF-10 THEN PRINT'" P
oor...give up!"
3050PROC_SPACE
3060ENDPROC
3070:
3080DEFPROC_AGAIN
3090CLS
3100PRINT''''"Do you want another go?"
3110PRINT'"Type Y or N"
3120PRINT'"then press RETURN. > ";
3130*FX15,0
3140a$=GET$
3150:
3160REM SET UP ONLY FOR Y/N
3170:
3180N%=1
3190IF a$="Y" OR a$="y" THEN N%=0
3200IF a$="N" OR a$="n" THEN N%=-1
3210IF N%=1 THEN3140
3220PRINTa$;
3230:
3240REM WAIT FOR RETURN
3250:
3260a$=GET$:IFa$<>CHR$(13)THEN3260
3270:
3280REM N% SET TO 0 FOR Y
3290REM OR -1 FOR N
3300:
3310ENDPROC
3320:
3330DEFPROC_WIN
3340:
3350REM GENERATE WIN SOUND
3360:
3370SOUND&0010,1,100,30
3380:
3390REM INCREMENT NO OF HITS
3400:
3410H=H+1
3420:
3430REM IMMEDIATE SCORE UPDATE
3440:
3450PROC_Flight_Info
3460ENDPROC
>
```





Program 7. Throbbing colours

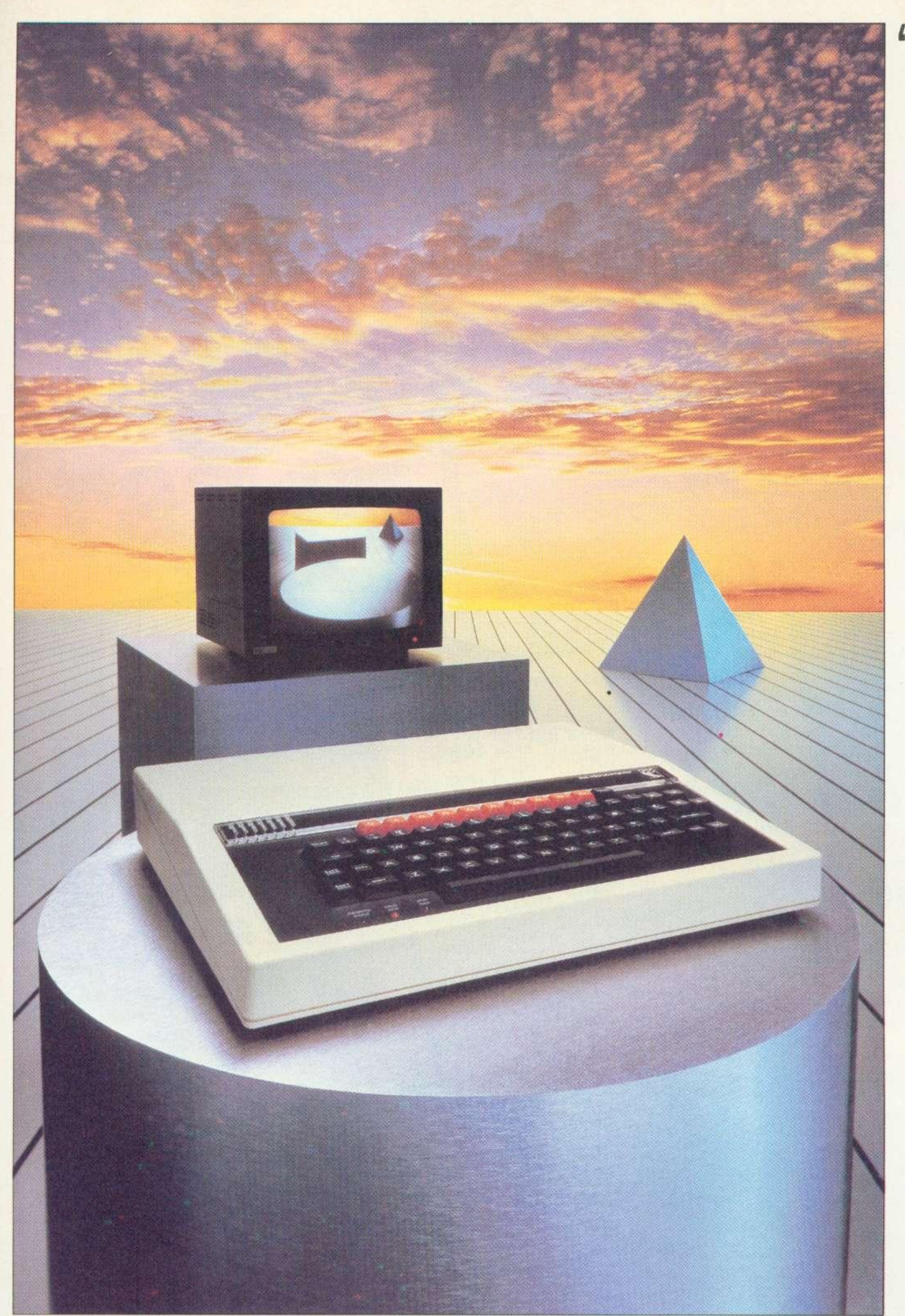
```
100REM THROBBING COLOURS
110:
120REM START WITH COLOUR 16
130:
140COL=16
150:
160REM USE MODE 2 FOR MAX COLOURS
170:
180MODE2
190REM BACKGROUND TEXT COLOUR IS RED
200COLOUR129:CLS
210:
220REM DRAW 3 MULTICOLOURED CIRCLES
230REM WHICH INTERLOCK
240REM USE THE FLASHING COLOURS
250:
260REPEAT
270COL=COL-1
280GCOL 0,COL
290PROC_CIRCLE(312,512,COL*20,"F")
300PROC_CIRCLE(612,512,COL*20,"F")
310PROC_CIRCLE(912,512,COL*20,"F")
320UNTIL COL =9
330:
340END
9300DEFPROC_CIRCLE(X,Y,R,F$)
9305LOCAL Z
9310 IF F$="F"THEN 9335
9315MOVEX+R,Y
9320 FOR Z = 2*PI TO -.2 STEP-.2
9325DRAW X+(R*COS(Z)),Y+(R*SIN(Z))
9330NEXT:ENDPROC
9335 MOVEX,Y
9340DRAWX+R,Y
9345 FOR Z = 2*PI TO -.2 STEP-.2
9350PLOT85,X+(R*COS(Z)),Y+(R*SIN(Z))
9355MOVEX,Y
9360NEXT:ENDPROC
```



```
100REM ADVERTISING BOARD
340REM NOW POP NAME INTO A$
350REM NO MORE THAN 14 CHARACTERS
360:
370A$="HAYES DECOR"
380:
390REM CYCLE FOREVER THROUGH THE ADS
400:
410REPEAT
420:
430REM GET PRODUCT INTO B$
440:
450READ B$
460:
470REM IF NO MORE GOODS - RESTART
480:
490IF B$ ="END" THEN RESTORE:GOTO450
500:
510REM NOW GET COST
520:
530READ C$
540:
550REM SET COLOURS:-
560REM FORGROUND FLASH BLUE/YELLOW
570REM BACKGROUND FLASH RED/CYAN
580REM THIS FITS IN WITH
590REM COLOUR OF LAST CIRCLE
600:
610COLOUR11:COLOUR137
620:
630REM NOW CENTRE TEXT ON SCREEN
640:
650PRINTTAB((10-LEN(A$)/2),13);A$
660PRINTTAB((10-LEN(B$)/2),15);B$
670PRINTTAB((10-LEN(C$)/2),17);C$
680:
690REM 10 SECS PER ADVERT
700:
710WT=TIME+1000:REPEAT UNTIL TIME>WT
720:
730REM RUB OUT TEXT FOR NEXT AD.
740:
750PRINTTAB(2,15),STRING$(15," ")
760PRINTTAB(3,17),STRING$(14," ")
770:
780REM NEVER STOP
790:
800UNTIL FALSE
810:
820REM DATA IN PAIRS...
830REM PRODUCT NAME THEN COST
840REM WITH 'END' AT THE END
850:
860DATA "BATHROOM SUITES"
870DATA"FROM `70"
880DATA"COPPER PIPING"
890DATA"97p per Foot"
900DATA"GUTTERING"
910DATA"80p per Metre"
920DATA"WASH BASINS"
930DATA"FROM `25"
940DATA"GARDEN TOOLS"
950DATA"Low Prices"
960DATA END
```

Program 8. Uses program 7 as background for an advertising display

Next month Joe Telford's tips sound a festive note



'Whether your interests lie in business, educational, scientific, control or games applications, this system provides a possibility for expansion which is unparalleled in any other machine available at present,' comments Paul Beverley in the July 1982 edition of *Personal Computer World*.

The BBC Microcomputer can genuinely claim to satisfy the needs of novice and expert alike. It is a fast, powerful system generating high resolution colour graphics and which can synthesise music and speech. The keyboard uses a conventional layout and electric typewriter 'feel.'

You can connect directly* to cassette recorder, domestic television, video monitor. disc drives, printers (dot matrix and daisy wheel) and paddles. Interfaces include RS423, inter-operable with RS232C equipment, and Centronics. There is an 8-bit user port and 1MHz buffered extension bus for a direct link to Prestel and Teletext adaptors and many other expansion units. The Econet system allows numerous machines to share the use of expensive disc drives and printers.

BASIC is used, but plug-in ROM options will allow instant access to other high level languages (including Pascal, FORTH and LISP) and to word processing software.

A feature of the BBC Microcomputer which has attracted widespread interest is the Tube, a design registered by Acorn Computers. The Tube is unique to the BBC Microcomputer and greatly enhances the expandability of the system by providing, via a high speed data channel for the addition of a second processor. A 3MHz 6502 with 64K of RAM will double processing speed; a Z80 extension will make it fully CP/M** compatible.

The BBC Microcomputer is also at the heart of a massive computer education programme. The government has recommended it for use in both primary and secondary schools. The BBC Computer Literacy Project includes two series of television programmes on the use and applications of computers.

There are two versions of the computer. Model A, at £299, offers 16K of RAM and Model B at £399 has 32K of RAM.

For technical specification and order form, send stamped addressed envelope to P.O. Box 7, London W3 6XJ and for details of your nearest stockist ring 01-200 0200.

Broader horizons

BBC Model B Microcomputers are available for immediate despatch by courier to your door.

If you would like to take advantage of our special delivery offer just fill in this coupon, but remember this only applies to the Model B, and only in the UK. Any other items have to be ordered on the standard order form or from a BBC Computer Stockist.

BBC Microcomputer System Offer
c/o Vector Marketing
Dennington Estate
Wellingborough
Northamptonshire NN8 2RL

Please send me by courier______BBC Model B Microcomputer(s) at £399 including VAT and delivery.

Cheque enclosed for £______
payable to BBC Microcomputer System readers a/c or charge
ACCESS BARCLAYCARD

Signed______
Name______
Address______
______
Postal Code______

Co Reg No 1403810 VAT No. 215400220

The BBC Microcomputer System

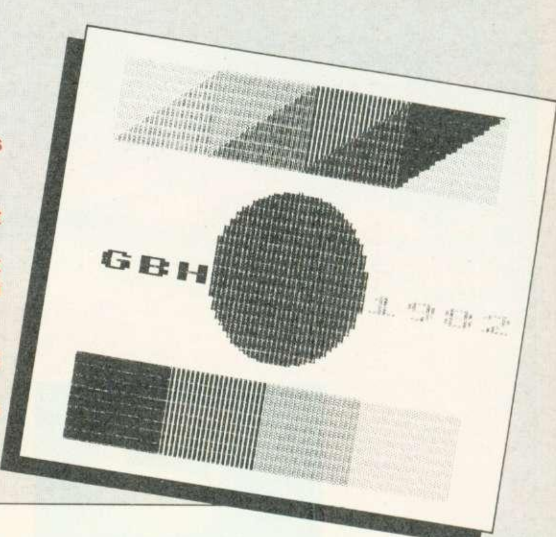
BBC Microcomputer System Offer, c/o Vector Marketing, Dennington Estate, Wellingborough, Northamptonshire NN8 2RL.

*Model A has a limited range of interfaces but can be upgraded to meet Model B specification.
**CP/M is a registered trade mark of Digital Research.

The BBC Microcomputer is designed, produced and distributed in the UK by Acorn Computers Limited.



EPSON MX80



In the October issue of *Acorn User*, George Hill explained the workings of the Epson MX80 F/T2 printer, and gave listings of single colour and two-tone dumps from the BBC micro.

Here is the final listing, to print a seven-tone picture, as in the 'testcard' shown on the right. The program is in Basic, but uses the fast VDU drivers to reproduce a complete screen in less than 30 minutes.

Having stored the program, numbered from 1000, proceed as follows:

- LOAD your graphics program
- RENUMBER it to ensure that it ends before line 1000
- PRINT~TOP-2 (this returns a hex number, "hexnumber", which you now use to merge the programs)
- *LOAD "BITPRINTn" hexnumber
- Type OLD to reset the BASIC pointers
- Replace the END of your graphics program by GOTO 1000
- RUN and hope!

```
1000 REM * * BITPRINT2 * * MODE 2 * *
1010  REM * Copyright G.B.Hill June 1982 *
1020  REM Seven tone picture dump for EPSON MX-80 F/T2

1030 REM ***variable declaration***
1040  REM X%,Y% screen coordinates
1050   REM P1,P2,P3,P4, stores for screen readout data
1060   DIM C(8,6) :REM C(I,J) used to define characters sent to printer
1070  FOR I=1 TO 6:C(0,I)=0:NEXT
1080  C(1,1)=0:C(1,2)=2:C(1,3)=0:C(1,4)=0:C(1,5)=1:C(1,6)=0
1090  C(2,1)=0:C(2,2)=1:C(2,3)=2:C(2,4)=1:C(2,5)=2:C(2,6)=0
1100   C(3,1)=1:C(3,2)=2:C(3,3)=2:C(3,4)=1:C(3,5)=1:C(3,6)=0
1110    C(4,1)=0:C(4,2)=3:C(4,3)=1:C(4,4)=3:C(4,5)=0:C(4,6)=3
1120  C(5,1)=0:C(5,2)=3:C(5,3)=3:C(5,4)=3:C(5,5)=3:C(5,6)=0
1130  C(6;1)=3:C(6,2)=3:C(6,3)=1:C(6,4)=2:C(6,5)=3:C(6,6)=3
1140  FOR I=1 TO 6:C(7,I)=3:NEXT
1150 REM set up printer,clear print buffer,restore screen 0
1160 PROCPRINTER
1170  REM scan screen and send data to printer
1180  PROCSCAN
1190  REM finishing routine
1200   PROCDONE
1210  END
1220  DEFPROCPRINTER
1230  MOVE 0,0 :REM zero graphics cursor
1240  *FX 5,2
1250  REM select RS423
1260  *FX 8,4
1270  REM 1200 baud
1280  VDU2:PRINT:VDU3.1,10,1,10,1,10 :REM clear print buffer
1290  VDU1,27,1,65,1,8 :REM Linefeed setting,ESC,A,8
1300  ENDPROC
1310  DEFPROCSCAN
1320   FOR Y%=1023 TO 0 STEP -16
1330    VDU1,27,1,76,1,192,1,3
1340        FOR X%=0 TO 1279 STEP 8
1350      P1=0:P2=0:P3=0:P4=0
1360                 P1=POINT(X%,Y%-12):P1=P1+8:P1=P1 MOD 8
1370                 P2=POINT(X%,Y%-8):P2=P2+8:P2=P2 MOD 8
1380                 P3=POINT(X%,Y%-4):P3=P3+8:P3=P3 MOD 8
1390                 P4=POINT(X%,Y%):P4=P4+8:P4=P4 MOD 8
1400           FOR J%=1 TO 6
1410         VDU1,(C(P1,J%)+4*C(P2,J%)+16*C(P3,J%)+64*C(P4,J%))
1420           NEXT
1430       NEXT
1440    VDU1,10 :REM Linefeed
1450     NEXT
1460  ENDPROC
1470   DEFPROCDONE
1480  VDU1,27,1,50 :REM Normal linefeed
1490  VDU 1,27,1,70 :REM Cancel condensed characters
1500  VDU1,12,1,7 :REM formfeed and beep
1510  VDU5
1520    PLOT 4,200,224
1530   PRINT"Picture complete."
1540   PLOT 4,560,160
1550   PRINT"BYE"
1560    VDU4,26 :REM separate graphics cursor, restore windows
1570   VDU31,0,31 :REM text cursor to bottom of screen
1580  ENDPROC
```





Light Pen

A complete program with plug-in light pen that will draw on the screen.
Ideal for games, graphics origination and commercial uses.
Fabulous value at **£30** plus VAT plus £1 p & p.

Graphics Tablet

Transfer your technical drawings or designs from pad to screen in seconds, and in colour.
Size 30" square.
Measures distances and areas and moves drawings on screen.
Only **£75** plus VAT plus £4.50 p & p.

128K Ram Pack

Simple plug-in memory expansion board, addressable for your own programs.
Introductory offer **£230** plus VAT plus £1 p & p.

BBC HARDWARE

Model 'A'
£299 (+ £7 p&p)
16K RAM 32K ROM. Full colour, high resolution graphics.
£399 (+ £7 p&p)
Model 'B' 32K RAM 32K ROM 16 Colour Graphics
£69
Tablet.
£130
BBC Disk Interface.
£265 (+ £2.50 p&p)
BBC 'B' Upgrade kit.
BBC Single Disk Drive 100K bytes.
£270 (+ £9.50 courier)
14" RGB Colour Monitor (as used in the BBC computer programme).
£95 (+ £6.50 courier)
12" Green Monitor.
£13 p. pair (+ £1 p&p)
BBC Joystick.
£288 (+ £4.50 p&p)
Acorn GP 100A Printer.
Epson MX80 FT111 Printer New model, dot matrix, high res. graphics 80 or 132 char's per line.
£399 (+ £4.50 p&p)
£15 (+ £1 p&p)
Printer Cable (parallel).

Dealers and overseas distributors enquiries welcomed.

Mail or telephone order,
Barclaycard or Access Credit card accepted.



MICRO MANAGEMENT

Micro Management, 32 Princes Street, IPSWICH. Telephone 59181



Scouse micros

The *BBC Microgroup Liverpool* holds meetings on the third Thursday of each month at Stem, 117 Grove Street, Liverpool (by the corner with Myrtle St) from 7.30 to 9.30 pm. Acorn Atoms are welcome. The group is affiliated to the *Merseyside Microcomputer Group c/o Fred Shaw, 14 Albany Avenue, Eccleston Park, Prescot, Merseyside, L34 2QW, to whom enquiries should be vectored.*

Preston Poly

The *Preston Area BBC Micro User Group* has recently been set up and aims to start a software library and a newsletter. Members' software will be evaluated by the club!

Meetings take place at Preston Polytechnic in room F2 at the moment, although this may change. The next meeting is 24 November. Anybody wishing to contact the group should send for details from: Mr D. Coulter, Preston Area BBC Microcomputer User Group, 8 Briar Grove, Ingol, Preston PR2 3UR

Micros for leisure

Caterham Leisure Centre has started a Computer Club. Meetings are held on Thursday nghts at 8.00 pm. The centre has a model B BBC computer available. For details ring the centre manager Mr M. Goldsbrough on Caterham 48304 or Mr. J. Hodges on Caterham 43316. The address is Godstone Road, Caterham, Surrey CR3 6RE.

Brum Beeb radio

Recently the *West Midlands Computer Group* was established with a monthly magazine – Databus. Meetings are being planned, which will be held in the centre of Birmingham and the group is to be represented on a monthly radio programme to be broadcast on Radio WM – the local BBC station.

For details, contact the membership secretary, 12 Apsley Road, Oldbury, West Midlands B68 0QZ.

Databus is edited by Mr D.J. Thompson at 130 Golden Cross Lane, Catshill, Bromsgove, Worcs.

Anybody else out there? Contact Acorn User, 53 Bedford Square, London WC1

CLUB CONTACTS

- Mr D. Coulter
  Preston BBC User Group
  8 Briar Grove
  Ingol
  Preston PR2 3UR
- Mr D.L. Evans
  23 Hitchin Road,
  Henlow Camp
  Bedfordshire
- Mr J. Price
  Bedford House
  27-28 St George's Road
  Brighton
  Sussex
- Mr J. Craig
  National BBC User Group
  40 Mount Pleasant Avenue
  Wells,
  Somerset BA5 2JQ
- Mr P. Frost
  Atom Users Group
  18 Frankwell Drive
  Potters Green
  Coventry CV2 2FB
- Mr R. Luff
  Kingbee
  54 Arlington Close
  Kingswinford
  West Midlands
- Steve White
  Atom/BBC User Group
  c/o Superior Systems Ltd
  178 West Street
  Sheffield
  Tel: (0742) 755005
- Mr C. Rutter
  Medway Atom Users Club
  St John Fisher School
  Ordnance Street
  Chatham
  Kent
- Mr B. Carroll
  The Cottage, 42 Manor Road
  Aldershot GU11 3DG
- **Beebug**
  374 Wandsworth Road
  London SW8 4TE
- Richard Green
  Muse
  22 Tennyson Avenue
  Hull HU5 3TW
- **West Midlands Computer Group**
  12 Apsley Road
  Oldbury
  West Midlands B68 0QZ
- Mr J. Ashurst
  Acorn Computer Users Group
  Abraham Moss Centre
  Crescent Road
  Manchester 8
- Rupert Steele
  Amateur Computer Club
  St John's College
  Oxford OX1 3JP
- **Liverpool BBC Microgroup**
  c/o Fred Shaw
  14 Albany Avenue
  Eccleston Park
  Prescot
  Merseyside L34 2QW
- Robin Bradbeer
  Association of London Computer Clubs
  Polytechnic of North London
  Holloway
  London N7 8DB
- Mr T.G. Meredith
  Acorn Atom User Group
  Sheerwater,
  Yealm View Road
  Newton Ferrers
  South Devon
- Mr C.M. Rutter
  Manchester Atom Users Group
  3 Leopold Avenue
  Withington
  Manchester M20 8JG
- **Muse** (for teachers)
  Freepost
  Bromsgrove
  Worcs B61 7BR
- N.P. (Bazyle) Butcher
  Harrow Computer Group
  16 St Peter's Close
  Bushey Heath
  Watford WD2 3LG
- Mr P. Beverley
  Norwich Area Acorn User Group
  Room 12a, Norwich City College
  Ipswich Road
  Norwich NR2 2LJ
- Mr M. Christiansen
  BBC Users Group
  Marienlystveien - Stavne
  N-7000 Trondheim
  Norway
- Paul Barbour
  Laserbug
  4 Station Bridge
  Woodgrange Road
  London E7 0NE
- Peter Smith
  Fareham and Portchester Amateur Computer Club
  23 Sandy Close
  Petersfield
  Hants.
- **Computer Club**
  Caterham Leisure Centre
  Godstone Road
  Caterham
  Surrey CR3 6RE
  Tel: Caterham 48304/43316


PROGRAM POWER MICRO POWER
Attention all ATOM Owners !!
Announcing the Software
SPECTACULAR
from
PROGRAM
POWER
TWO NEW 4K ROMS
MACHINE CODE MONITOR/ DISASSEMBLER
*SINGLE STEP with REGISTER DISPLAY
*BREAKPOINT handling routines
*Tabulation Memory Dump with 'on-screen' Editing (printer option)
*Memory Display—Hex, ASCII, Screen Code, Decimal & Binary
*SEARCH—Hex or ASCII string
*INTELLIGENT COPY
*COMPARISON of two memory blocks
*DISASSEMBLER (full feature)
*INCREDIBLE PRICE—ONLY £14.50
SUPER TOOLBOX
*SUPER-FAST Cassette Operation System with error checking, verify and append routines in addition to original 300/1200 baud.
*TOOLKIT functions—AUTO, RENUMBER, HELP, FIND, SPACE, APPEND, MEM, VAR, LVAR, DUMP, ZERO, HEX, IHEX, DELETE.
OTHER FEATURES—Flashing Cursor, 2-key Rollover, Repeat Key, Fast Repeat.
*SUPER LOW PRICE— ONLY £14.50.
PROGRAMMERS TOOLBOX
(PACKED 4K EPROM) £24.50
★1200 BAUD CASSETTE OPERATING SYSTEM
★ VISIBLE LOAD & SAVE
★29 additional BASIC & TOOLKIT COMMANDS
PROGRAMMERS TOOLBOX NOW ONLY £17.50
ROM SELECTOR BOARD
Suffering from congestion? Switch between up to 4 ROMS located at Hex A000. Assembled and tested unit with compact professionally produced PCB and good quality components. Fits easily into the Utility Socket (IC 24). The additional ROMS above make having a selector board at only £19.50 even more essential!
MACHINE CODE PROGRAM PRICES ARE TUMBLING!!
WRITTEN ANY PROGRAMS! WE PAY 20% ROYALTIES FOR ATOM, NASCOM & BBC MICRO PROGRAMS
ABCDEFGH
YOUR MOVE
SCORE HIGH SCORE
32990 60090
Chess (12K) Improved graphics, plays black or white, mid game level changes, look ahead up to eight moves, offensive, normal and defensive play. Ten sub-levels, castling, 'En passant' by player. Rejects illegal moves. Take back moves and action replay with take over. Set up problem games. £6.95.
Hyperfire (12K colour) Aliens have over-run six planets. Object to clear them. Super high-speed action. Four defender ships. Hit alien fuel dumps and alien ships. Aircraft radar shows placement of approaching ships. Thruster control—including diagonal movement. Best sound effects yet! £6.95
Dog Fight (12K) Exciting, two player combat program—in HI RES Graphics. Defend your aircraft carriers and attack your opponents. Engage the enemy fighters in a battle to the death £5.95
ASTROBIRDS (12K) NEW version of this incredible GALAXIAN type game. Fantastic sound effects. Screaming missiles & swooping bird men. NOW DOUBLE SPEED OPTION. £5.95.
ADVENTURE (12K-NOT M/C) ALL THE INTRIGUE AND FRUSTRATION OF A MAIN-FRAME ADVENTURE! Great skill and imagination are required to play this excellent game and you may still never exhaust all the possibilities £6.95.
Cowboy Shoot-Out (12K) Full feature, two player, arcade shooting game. Superb graphics and sound £5.95
Stock-Car (12K) HI RES. Two Player contest around any of 16 different race tracks. Steering, acceleration and braking controls. Set the level of difficulty for driver and choose the no. of laps £5.95
SPACE FIGHTER (12K COLOUR) Super High speed "Defender" game. 5 types of intelligent aliens. Repeating laser cannon, smart bombs, hi-score, rankings, bonus points, 6 skill levels. Exciting sound effects £6.95.
INVADER FORCE (12K) Terrific version of 'Space Invader' 4 types of invaders, mother ship, great sound, hi-score, 6 skill levels £5.95
ATOMSTORE (12K-NOT M/C) Database filing & printing program, excellent value at only £6.95
PLEASE NOTE: WE ARE AUTHORISED DEALERS FOR ACORN ATOM, BBC MICRO & DRAGON 32
ATOM SPECIAL OFFER
Deduct £1 per cassette when ordering two or more.
WE Guarantee THAT ALL OUR ADVERTISED PROGRAMS HAVE BEEN COMPLETED AND ARE READILY AVAILABLE
Please add 55p order P & P + VAT at 15%
MICRO POWER LTD.
8/8A REGENT STREET,
CHAPEL ALLERTON
LEEDS LS7 4PE.
Tel. (0532) 683186
VISA
PROGRAM MICRO POWER

SOFTWARE FOR THE B.B.C. MICRO FROM....

COMPUTER CONCEPTS





LOGO II

MODEL B

The first implementation on the BBC Micro of the graphics language LOGO.
This language is now very popular in American schools as it is an ideal educational program.
It can graphically demonstrate the ideas of defined procedures, sub routines, loops and even recursive programming. This program will give an excellent introduction to LOGO language, for young and old alike!

EDUCATIONAL

£10.00 +VAT



HITCH-HIKER

MODEL B

'Hitch-Hiker' is a great adventure game based on characters from the book 'Hitch-Hikers Guide to the Galaxy'. A fascinating game to test your skill and wits whilst trying to collect five objects scattered around the Universe located in such places as the 'Restaurant at the end of the Universe', 'Arthur Dents House' and Belteguese Spacedrome'. Directions can be found in the clues. An intriguing game that can create hours of fun.

£5.80 +VAT

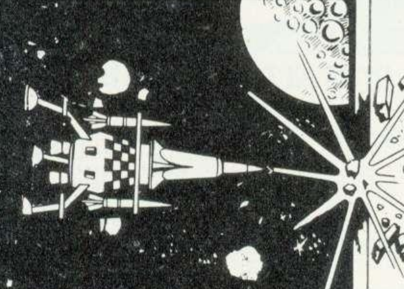


SPACE HAWKS

MODEL B

A great arcade type game similar to the popular 'space invaders'. A game of skill and speed. Armed with photon bolts to defend yourself — try to beat off the attack of the invaders and flapping space hawks. Makes full use of high resolution graphics.

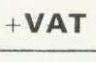
£7.80 +VAT



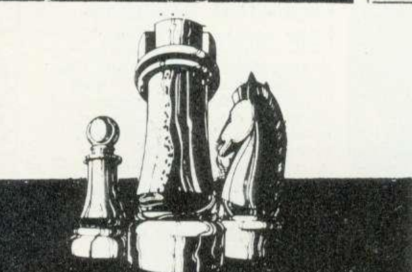
ASTEROID BELT

MODEL A OR B

A great new space game pratically identical to the arcade original. You are stuck in the middle of a cloud of asteroids against which you have no protection — your only chance of survival is to destroy the asteroids with your photon bolts. Manoevring can be achieved by rotating and thrusting. As a last resort you can 'hiper-space', immediately transporting you to a random position.
An inspired piece of machine code programming producing one of the most exciting games around.



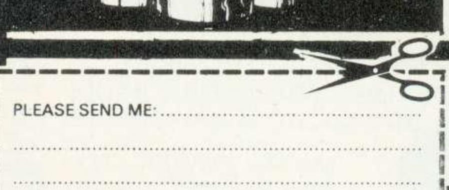
£7.80 +VAT



CHESS MODEL B

Another machine code program for the Model B. This game has a host of facilities.
An enormous range of skill levels — you can alter such parameters as the number of moves that the program looks ahead, the speed, and sub-levels, etc.
The computer can act as an umpire for two players. It can even play against itself.
This program uses high resolution colour graphics to display the board and its pieces. An excellent introduction to beginners though it still gives the experienced player a real challenge.

£10.00 +VAT



PLEASE SEND ME: ....
....
....

I enclose £ ....

NAME ....

ADDRESS ....
....
....

in our opinion the best B.B.C. software in town!



Dept AC4

16 Wayside, Chipperfield, Herts, WD4 9JJ. tel (09277) 69727

Send a S.A.E. to get the latest details of our software.



GRAPHICS ROUTINES

GRAPHICS ROUTINES
GRAPHICS ROUTINES

Up, down, left, right, change colour – these machine code routines by Paul Beverley make it all possible. Just type them in and follow the instructions

This article describes a series of machine code routines for manipulating Atom graphics – both colour and black and white. They provide rotations (up, down, left and right), inversions (up/down and left/right), global colour changes, and block moves of memory for exchanging views.

Machine code has the advantage over Basic because it lets you work fast enough to achieve reasonable animation effects – but don't panic if you have never used machine code before. To use these routines, you only need to be able to type them in and follow the instructions!

The routines are written in what are called assembler mnemonics (program 1). In this form they are vaguely readable, but they occupy over 3½k of memory. Once run, the assembler routine is turned into the machine code which the computer understands, and only uses just over ½k (589 bytes) leaving a reasonable space for the program which is going to use the routines.

However, there is no need to load and run this every time you want to use the routines. They can be saved in machine code form by saying

```
*SAVE "M-CODE" 39B0 3BFF
```

and then re-loaded when you want to use them again by saying

```
*LOAD "M-CODE"
```

Program 2 complements the explanations of the routines by showing how they can be used. It would take too long to explain fully the operation of this program, but it is laid out in sections which should

Table 1. Summary of single key commands available after the initial display in zero mode of program 1

- A Set the starting page of the data to be worked on.
- C Clear one of the graphics modes. Press one of keys 0 to 8 to specify mode (as shown in table 2).
- D Move down. (SHIFT D for continuous motion until CTRL is pressed.)
- E Exchange pictures. Press either key 0 to allow input of the page addresses from the keyboard, or keys 1 to 6 to select two of the six 1k pictures (mode 1a) to be exchanged.
- F Repeats the zero mode demonstration which was shown at the start of the program.
  Change graphics mode without clearing. Press one of keys 0 to 8.
- H Change height of picture being worked on by routines.
- I Invert the picture.
- L Move left. (SHIFT L for continuous motion until CTRL is pressed.)
- M Mirror image = lateral inversion = left-right inversion.
- P An example of animated graphics.
- Q Colour changes – press key 1, 2 or 3.
- R Move right. (SHIFT R for continuous motion.)
- S Save a view into another area of memory – uses same format as exchange routine.
- T Draw a train.
- U Move upwards. (SHIFT U for continuous motion.)
- W Change width of picture.
- X Horizontal colour bars in mode 1a.
- Y Vertical colour bars in mode 1a.
- Z Coloured squares in mode 1a.

SHIFT S – automatically standardises the height and width of the current graphics mode.

Table 2. Parameters associated with graphics modes

Graphics mode	?#B000 (=?N)	Height (bytes)	Width (bytes)	Memory (k) (x 1024)	Memory (pages) (x 256)	Number used by program
0	0	16	32	½	2	0
1	#30	64	16	1	4	2
2	#70	96	16	1½	6	4
3	#B0	192	16	3	12	6
4	#F0	192	32	6	24	8
1a	#10	64	16	1	4	1
2a	#50	64	32	2	8	3
3a	#90	96	32	3	12	5
4a	#D0	192	32	6	24	7



aid readability. The operating instructions are given in table 1.

The Atom has more different graphics modes than a model B BBC microcomputer – and this can cause problems. It has a total of nine – five in black and white and four in colour.This means you have to provide information on the current mode for the routines to refer to.

The routines could find out directly which is the current mode of graphics by reading memory location #B000 – port A of the 8255 interface adaptor. As can be seen from table 2, it is the upper four bits of this port which determine the mode.

However, to make the routines more flexible, it was decided to use a location in zero page (# 87) to store this number. In other words, to use the current mode, you copy #B000 into #87 (?#87 = ?#B000) and the routines then look at #87.

So by changing the contents of #87, the routine can perform as if the Atom were in a different mode altogether.

Similarly, various other parameters are stored in zero page. The number of horizontal lines on the screen (16 to 192) is stored at #85, and the width of the screen in terms of bytes (16 or 32) is stored at #84.

For routines which move whole chunks of memory around, the start addresses of the two areas of memory are stored at #80, #81 and #82, #83, in the usual 6502 notation – ie low byte followed by high byte.

But since we will only be dealing with whole multiples of 256 bytes, the low byte will always start at zero. The high bytes (#81 and #83) therefore contain the initial page numbers of the blocks of memory being manipulated.

The other parameter necessary for some of the routines is stored at #88, and is the initial page number of the memory on which the routine is to act. This would normally be #80 since the active screen memory always starts at #8000, but sometimes you might want to manipulate a different part of memory and exchange it with the current screen memory.

Program 1

```
10 T=#80;F=#82;W=#84;H=#85;Q=#86;N=#87;B=#88;V=#89;K=#90
20 DIM LL37;FORJ=0TO37;LLJ=-1;N.
30 P.$21;FORJ=1TO2;P=#39B0;[
40\
50\COLOUR CHANGES  ?Q=#FF or 55 or AA
60\
70:LL0 JSRLL33;JSRLL36
80:LL1 LDA(T),Y;EORQ;STA(T),Y;DEY;BNELL1
90      INCT+1;DEX;BNELL1;RTS
100\
110\BLOCK EXCHANGES (SET F?1 & T?1)
120\
130:LL2 LDY@0;STYF;STYT;JSRLL36
140:LL3 LDA(F),Y;PHA;LDA(T),Y;STA(F),Y;PLA;STA(T),Y
150      INY;BNELL3;INCF+1;INCT+1;DEX;BNELL3;RTS
160\
170\LOAD/STORE (SET T?1 & F?1)
180\
190:LL4 LDY@0;STYF;STYT;JSRLL36
200:LL5 LDA(F),Y;STA(T),Y;INY;BNELL5
210      INCF+1;INCT+1;DEX;BNELL5;RTS
220\
230\ROTATE RIGHT
240\
250:LL6 JSRLL33;LDAH;STAQ
260:LL7 LDYW;DEY;LDA(T),Y;PHA;DEY
270:LL8 LDA(T),Y;INY;STA(T),Y;DEY;DEY;BPLLL8
280      INY;PLA;STA(T),Y;JSRLL34;BNELL7;RTS
290\
300\ROTATE LEFT
310\
320:LL9 JSRLL33;LDAH;STAQ
330:LL10LDY@0;LDA(T),Y;PHA;INY
340:LL11LDA(T),Y;DEY;STA(T),Y;INY;INY;TYA;CMPW;BNELL11
350      DEY;PLA;STA(T),Y;JSRLL34;BNELL10;RTS
360\
370\LATERAL INVERSION
380\
390:LL12JSRLL33;LDAH;STAQ;LDYW;DEY;STYV
```

```
400:LL13LDAV;LSRA;TAY
410:LL14LDA(T),Y;JSRLL15;TAX;TYA;EORV;TAY
420      LDA(T),Y;JSRLL15;PHA;TXA;STA(T),Y;TYA;EORV;TAY
430      PLA;STA(T),Y;DEY;BPLLL14
440      JSRLL34;BNELL13;RTS
450\COLOUR OR B & W?
460:LL15PHA;LDAN;BEQLL17;ASLA;ASLA;BMILL16\ COLOUR OR B&W?
470\INVERT A COLOUR BYTE
480      PLA;RORA;RORA;ROLK;ROLA;ROLK;RORA
490      RORA;RORA;ROLK;ROLA;ROLK;RORA
500      RORA;RORA;ROLK;ROLA;ROLK;RORA
510      RORA;RORA;ROLK;ROLA;ROLK;RORA;LDAK;RTS
520\INVERT A B&W BYTE
530:LL16PLA;RORA;ROLK;RORA;ROLK;RORA;ROLK;RORA;ROLK
540      RORA;ROLK;RORA;ROLK;RORA;ROLK;RORA;ROLK;LDAK;RTS
550\IN ZERO MODE - IS IT A GRAPHICS CHARACTER?
560:LL17PLA;ROLA;BMILL18;RORA;RTS
570\IF SO INVERT IT
580:LL18RORA;RORA;ROLK;RORA;PHP;ASLA;ASLA;PLP;RORA;RORK;RORA
590      RORA;ROLK;RORA;PHP;ASLA;ASLA;PLP;RORA;RORK;RORA
600      RORA;ROLK;RORA;PHP;ASLA;ASLA;PLP;RORA;RORK;RORA
610      RORA;PHP;ASLA;PLP;RORA;RORA;PHP;ASLA;PLP;RORA;RTS
620\
630\UP DOWN INVERT
640\
650:LL19JSRLL33;LDA@0;SEC;SBCW;
   STAF;JSRLL36;DEX;STXQ
660      LDAB;CLC;ADCQ;STAF+1;
   LDAH;LSRA;STAQ
670:LL20LDYW;DEY
680:LL21LDA(T),Y;JSRLL23;TAX;LDA
   (F),Y;JSRLL23;STA(T),Y;TXA
690      STA(F),Y;DEY;BPLLL21;
   LDAF;BNELL22;DECF+1
700:LL22SEC;SBCW;STAF;JSRLL34;
   BNELL20;RTS
710\ZERO MODE?
720:LL23PHA;LDAN;BEQLL24;PLA;RTS
730\GRAPHICS CHARACTER?
740:LL24PLA;ROLA;BMILL25;RORA;RTS
750\INVERT CHARACTER
760:LL25RORA;PHA;AND@3;ASLA;ASLA;
   ASLA;ASLA;STAK
770      PLA;PHA;AND@48;LSRA;LSRA;
   LSRA;LSRA;STAK+1
780      PLA;AND@#CC;ORAK;
   ORAK+1;RTS
```

```
790\
800\ROTATE UPWARDS
810\
820:LL26JSRLL33;STAF+1;LDYW;STYF;DEY
830:LL27LDA(T),Y;STAK,Y;DEY;BPLLL27;INY
840      JSRLL36;JSRLL5;LDXW;DEX;DEY;DECT+1
850:LL28LDA K,X;STA(T),Y;DEY;DEX;BPLLL28;RTS
860\
870\ROTATE DOWNWARDS
880\
890:LL29JSRLL36;DEX;STXQ;LDAB;CLC;ADCQ
900      STAT+1;STAF+1;LDY@0;STYT;STYF;DEY;LDXW;DEX
910:LL30LDA(T),Y;STAK,X;DEY;DEX;BPLLL30;LDAW;STAT;JSRLL36
920:LL31LDA(F),Y;STA(T),Y;DEY;BNELL31
930      LDA(F),Y;STA(T),Y;DEY;DECF+1;DECT+1;DEX;BNELL31
940      INCT+1;LDY@0;STYT;LDYW;DEY
950:LL32LDAK,Y;STA(T),Y;DEY;BPLLL32;RTS
960\
970\INITIALISE VARIABLES
980\
990:LL33LDAB;STAT+1;LDY@0;STYT;STYF;RTS
1000\
1010\MOVE DOWN TO NEXT LINE
1020\
1030:LL34LDAT;CLC;ADCW;STAT;BCCLL35;INCT+1
1040:LL35DECQ;RTS
1050\
1060\CALC. NO. OF BLOCKS
1070\
1080:LL36LDAW;ASLA;ASLA;STAV
1090      LDAH;LSRA;LSRA;LSRA;BITV;BMILL37;LSRA
1100:LL37TAX;RTS
1110];N.;P.$6
1120 @=1;P.$12"FOR ROUTINE...  LINK..  =#"&LL0"+"''
1130 P."COLOUR CHANGES          #"&LL0"    ",0'
1140 P."EXCHANGE VIEWS          #"&LL2"    ",(LL2-LL0)'
1150 P."LOAD/STORE VIEWS        #"&LL4"    ",(LL4-LL0)'
1160 P."ROTATE RIGHT            #"&LL6"    ",(LL6-LL0)'
1170 P."ROTATE LEFT             #"&LL9"    ",(LL9-LL0)'
1180 P."LEFT-RIGHT INVERT       #"&LL12"   ",(LL12-LL0)'
1190 P."UP-DOWN INVERT          #"&LL19"   ",(LL19-LL0)'
1200 P."ROTATE UPWARDS          #"&LL26"   ",(LL26-LL0)'
1210 P."ROTATE DOWNWARDS        #"&LL29"   ",(LL29-LL0)''
1220 P."ASSEMBLY ENDED AT       #"&P"   ",(P-LL0)'
1230 END
```



Table 3. Effects on different colours of EORing with different numbers

?#86=	change (and vice versa)
#FF	colour 0 to colour 3 colour 1 to colour 2
#AA	colour 0 to colour 2 colour 1 to colour 3
#55	colour 0 to colour 1 colour 2 to colour 3

An example of this is the animated graphics in the demonstration program where three different views of a train are shown on the screen in turn, each one being rotated to the left while in an area of memory not being displayed.

Having digested all this, let's look at the individual routines.

Colour changes: This routine simply goes through an area of memory doing an "exclusive OR" (a particular logical manipulation) on each byte with another parameter held in zero page (stored at #86). The three possible numbers to put at #86, and their effects, are shown in table 3.

Block exchanges: The size of the data blocks involved is first calculated (= ?#84 * ?#85 bytes) and the starting page numbers of each block have to be stored at #81 and #83.

Load/Store Data: This is set up in a similar way to the exchange routines, but here the contents of pages ?#81 and following are overwritten by the contents of pages ?#83 and following.

Rotations: Up, down, left, right. These four routines take ?#84 by ?#85 bytes of data, starting from page ?#88 and rotate them by one whole byte in the relevant direction.

Inversions: Lateral or up/down. These routines invert the view as specified by the same parameters used in the rotation routines, but they also have to check which mode of graphics is being used. The routines will be modified automatically according to whether the view is in colour or black and white, and if it is full graphics or, in zero mode, semi-graphics (pixels)/or text.

Program 2

```
10 T=#80;F=#82;N=#87;K=#91
20 B=#B000;?N=?B;F?6=T;A=#8000;H=#3FF
30 C=#39B0;E=C+21;S=C+51;R=C+75;L=C+108;M=C+143
40 I=C+321;U=C+429;D=C+469
50 DIMLL1,P-1;G=P;P.$21;[JSR#FFE3;STAK;RTS
60:LL0 LDA@T;STAT+1;LDA@T+4;STAT+3;JSRE;JSRL;JSRL;RTS
70:LL1 JSRLL0;LDAB+1;CMP@#FF;BEQLL1;RTS;]
80 P=#55;Z=4;Y=64
90 P.$6;G.a
100mLI.G
110 IF?K=65 GOS.i;IN."START FROM PAGE"X;F?6=X;GOS.j
120 IF?K=67 G.x
130 IF?K=68 LI.D
140 IF?K=69 GOS.n;LI.E;G.m
150 IF?K=70 G.a
160 IF?K=71 G.g
170 IF?K=72 GOS.i;IN."HEIGHT"X;F?3=X;GOS.j;G.m
180 IF?K=73 LI.I
190 IF?K=76 LI.L
200 IF?K=77 LI.M
210 IF?K=80 G.c
220 IF?K=81 LI.G;T?6=(?K%3+1)*P;LI.C;G.m
230 IF?K=82 LI.R
240 IF?K=83 GOS.n;LI.S;G.m
250 IF?K=84 GOS.t;G.m
260 IF?K=85 LI.U
270 IF?K=87 GOS.i;IN."WIDTH"X;F?2=X;GOS.j;G.m
280 IF?K=88 FORJ=0TOH;A?J=J/Y%Z*P;N.
290 IF?K=89 FORJ=0TOH;A?J=J%Z*P;N.
300 IF?K=90 G.z
310 IF?K=100 DO LI.D;U.B?1=191
320 IF?K=108 DO LI.L;U.B?1=191
330 IF?K=114 DO LI.R;U.B?1=191
340 IF?K=115 G.s
350 IF?K=117 DO LI.U;U.B?1=191
360 G.m
370
380 CHANGE GRAPHICS MODE
390
400gLI.G;J=?K-48;IFJ<0ORJ>8 P.$7;G.g
410 J=32*J-16;IFJ<0 J=0
420 ?B=J;?N=J;G.m
430
```

```
440 INPUT PAGE NUMBER
450
460nGOS.o;IF J>-1 F?1=T+4*J;GOS.o;T?1=T+4*J;R.
470eGOS.i;IN."FROM"J;F?1=J;P.$30"              "$30
480 IN."TO"J;T?1=J;GOS.j
490 IF T?1<#28 ORF?1<#28 P.$7;G.e
500 R.
510oLI.G;J=?K-49;IFJ<-1 OR J>5 P.$7;G.o
520 R.
530
540 MOVE IN & OUT OF ZERO MODE
550
560iFORJ=0TO25;K?J=A?J;N.;?B=0;P.$30;R.
570j?B=?N;FORJ=0TO25;A?J=K?J;N.;R.
580
590 DRAW A TRAIN
600
610tCOLOUR1;FORJ=0TO4;MOVE0,J;DRAW63,J;N.
620 COLOUR2;MOVE15,5;GOS.v;MOVE30,5;GOS.v
630 COLOUR3;FORJ=8TO16;MOVE13,J;DRAW37,J;N.
640 MOVE15,17;PLOT1,0,6;PLOT0,1,0;PLOT1,0,-6
650 FORJ=17TO24;MOVE28,J;DRAW37,J;N.
660 COLOUR1;FORJ=16TO20;MOVE31,J;PLOT1,3,0;N.;R.
670vPLOT1,3,0;PLOT0,1,1;PLOT1,-5,0;PLOT0,0,1;PLOT1,5,0;R.
680
690 SET HEIGHT & WIDTH OF GRAPHICS
700
710sJ=?B;?N=J;IFJ=0 F?2=32;F?3=16;G.m
720 IFJ<#50 F?2=16;F?3=64
730 IFJ=#70 F?2=16;F?3=96
740 IFJ=#B0 F?2=16;F?3=192
750 IFJ=#50 F?2=32;F?3=64
760 IFJ=#90 F?2=32;F?3=96
770 IFJ>#90 F?2=32;F?3=192
780 G.m
790
800 CLEAR DESIRED GRAPHICS MODE
810
820xLI.G;J=?K-48;IFJ<0ORJ>8P.$7;G.x
830 X=(J+1)/2;CLEARX;IFJ%2 COLOUR0
840 ?N=?B;G.m
```

```
850
860 SQUARES IN MODE 1A
870
880zFORJ=0TOH;IFJ%Y=0 X=X+1
890 A?J=(J+X)%Z*P;N.;G.m
900
910 ANIMATED TRAIN
920
930cCLEAR3;?B=16;?N=16;F?6=#84;F?2=16;F?3=64
940 GOS.t;COLOUR2
950 J=4;MOVE19,29;GOS.d
960 J=6;MOVE24,35;GOS.d
970 T?1=T;F?1=T+4;LINKE;LINKL
980 GOS.t;COLOUR2;J=3;MOVE17,27;GOS.d
990 J=5;MOVE21,32;GOS.d
1000 J=7;MOVE28,40;GOS.d
1010 DO LI.LL0;U.B?1<255;F?6=T;G.m
1020dFORX=1TOJ;PLOT1,J,0;PLOT0,-J,0;PLOT0,0,1;N.;R.
1030
1040 PICTURE IN MODE ZERO
1050
1060aP.$12;?#E1=0;CLEAR1;?B=0;?N=0;F?6=T
1070 P."@G>=K@@?<<<<@?<<<?@?<<<?@?K@@@@?"
1080 P."@8@@4/@?@@@@@?@@@?@?@@@?@?4/B@@?"
1090 P."?OOOO?@?@@@@@?@@@?@?OOO?@?@P=K@?"
1100 P."?@@@@?@?@@@@@?@@@?@?P=K@@?@@@4/?"
1110 P."?@@@@?@?OOOO@?OOO?@?@@4/@?@@@@P?"
1120 P."@@@@G>=K@@<<?<<@?<<<?@?/B@A?@@@"
1130 P."@@@@8@@4/@@@?@@@?@@@?@?P=O> ?@@@"
1140 P."@@@?OOOO?@@@?@@@?@@@?@?@@@@@?@@@"
1150 P."@@@?@@@@?@@@?@@@?@@@?@?@@@@@?@@@"
1160 P."@@@?@@@@?@@@?@@@?OOO?@?@@@@@?@@@"
1170 A?32=31;A?#DB=31;A?#E3=31
1180 F?2=32;F?3=16;T?6=64;LI.C;GOS.b
1190 FOR J=1TO5;WAIT;LI.D;N.;GOS.b
1200 WAIT;LI.I;GOS.b;WAIT;LI.M;GOS.b
1210 T?6=128;LI.C;GOS.b;WAIT;LI.I;GOS.b;WAIT;LI.M;GOS.b
1220 WAIT;LI.U;GOS.b;WAIT;LI.U;GOS.b
1230 FOR J=1TO128;LI.L;N.;P.$30"READY"
1240 FOR J=1TO59;T?6=128;WAIT;WAIT;WAIT;LI.C;N.;G.m
1250bFOR J=1TO60;WAIT;N.;R.
```



Do you want control of text size in all graphic modes?

Do you want text written up-side down or on its side?

Do you want a user-defined character in several colours?

If so, read on.

TEXT VARIATIONS

Although the *User Guide* states that only in mode 7 can you have double height characters, there is a simple routine which allows an increase in text size for all graphic modes from normal until a single letter fills the screen. This is particularly useful if you want a larger text size with the highest resolution graphics in mode 0.

And it works equally well with user-defined characters and allows you to use all the colours available in the mode you are in to within one character.

The following Basic routine shows how (program 1). It uses the function POINT to examine the letter to be magnified which is placed at TAB(0,0). This is converted to a graphics routine which draws the character in a larger size elsewhere on the screen.

This is sufficiently fast for headings and static displays but for a rapid display a short machine code programme is needed. In this the OSWORD call with A=9 should be used to examine the pixels that make up the printed character. The equivalent of the Basic PLOT routine for drawing the magnified characters by filling triangles is entered via

```
VDU25,85;X;Y
```

Returning to program 1, insert line

```
125 Y%= -Y%
```

This will produce text upsidedown! Now use

```
LIST 190,220
```

Use the copy facility to reverse the position of the X axis and Y axis in each statement. The text should now be written on its side from bottom to top. Simple routines thus allow any size of text to be placed in position on the screen.

A boon from abbreviations

Abbreviations can be a boon for loading, saving and running machine code programs. "*SAVE", for example, shortens to "*S." – though it must be followed by a filename and various addresses and so is not much of a saving.

Much more useful is knowing that "*LOAD" and "*RUN" can be shortened to "*L." and "*R.". And if you are using cassette files but do not want to specify the filename, there is no need to follow these commands with double quotation marks. In fact "*R." can be abbreviated still further to "*/".

The equivalent abbreviations for Basic programs are "LO.", "SA." and "CH." which, since they have to be followed by a filename, or at least double quotation marks, are hardly worth bothering with. Two other abbreviations worth knowing are "*E." for "*EXEC", and "*SP." for "*SPOOL". but both must have either a filename or a pair of quotes, except when "*SPOOL" is being used to close a spool file.

It is also handy to know that "*CAT" can be entered as "*.". This is useful in its own right, but especially so when you realise it can be used as an equivalent of "*MOTOR1" to switch on the cassette rewind motor. The equivalent of "*MOTOR0" is then just ESCAPE, since this turns the motor off again.

Program 1

```
10   MODE 7
20   INPUT TAB(0,5) "What mode do you want?" r

30   IF NOT (r=0 OR r=1 OR r=2 OR r=4 OR r=5)
     THEN PRINT "Not GRAPHICS MODE":GOTO20

40   MODEr
50   INPUT TAB(0,5) "How much bigger do you want
     text?"n
60   P% = 0 : Q% = 900-32*n
70   CLS
80   REPEAT
90   VDU30

100  R$ = GET$ : PRINT R$

110  FOR H% = 991 TO 1023 STEP 4
120  Y% = H% -991
«/)  FOR G% = 1 TO 16 STEP 2
140  X% = G% : M% = 2
150  IF r = 1 OR r= 4 THEN X% = G%*2: M% = 4
160  IF r = 2 OR r= 4 THEN X% = G% *2: M% = 4
160  IF r = 2 OR r = 5 THEN X% = G%*4 :M% = 8
170  A= POINT (X%, H%)

180  GCOL 0,A
190  MOVE (P% + (X%*n)), ((Y%*n)+Q%)
200  MOVE (P%+(X%*n)+(M%*n)),((Y%*n)+Q%)
210  PLOT85, (P%+(X%*n)),((Y%*n)+W%+4*n)
220  PLOT85,(P%+(X%*n)+(M%*n)),((Y%*n)+Q%+4*n)
230  NEXT G%
240  NEXT H%
250  P%=P%+X%*n
260  IF P%+X%*n>1280 THEN P% = 0:Q%=Q%-32*n

270  IF Q% > 1 THEN UNTIL FALSE

280  VDU31,0,31

290  FOR B% = 1 to n
300  VDU 10
310  NEXT B
320  Q% = Q% + 32*n
330  UNTIL FALSE
```

Notes

30 Checks value r is a valid graphics mode. If not valid returns to Line 20

40 Enters graphics mode

50 n will be the multiplication factor

60 P% locates character on X axis, Q% on Y axis. Q%=900-32*n ensures sufficient space above the starting point to draw the full height of the character

90 This is used to home cursor so that the next PRINT will place a character in TAB(0,0) – in Line 100

100 GET$ holds the programme until you type a key. If you type more than one key they will be stored in sequence until the programme returns for them.

110-160 Will locate in turn each pixel that makes up the character

140-160 This depends on which mode you are in. X% and H% are the X axis and Y axis reference for each pixel
X% and Y% are the axis references for each pixel
where the bottom left pixel is X%=1 and Y%=1

170 A now becomes the foreground colour of the pixel being examined.

180 Use the same colour as found in line 170

190-220 This routine draws the magnified pixel starting at position P%, Q%. The position references X% and Y% from lines 120 and 140 are multiplied by the magnification factor n from line 50

230-240 Repeat this process for each pixel in the printed character

250 Calculate where the next enlarged character should start.

260 If there is not enough room to draw the next character start a new line: move down enough to fit the height of the magnified character

270 If this new line is on the screen go to line 80. REPEAT to repeat the whole sequence for the next character

280 Move cursor to bottom left corner to initiate screen scroll. (NB. The *User Guide* suggests VDU31 should use graphic X and Y values. My computer EPROM 0.1 responds to TAB values)

290-310 Scroll the screen until the next character will fit

320 Reset value of Q%

330 Go to line 80. REPEAT, to repeat the whole sequence for the next character



DUMB TERMINALS

by Paul Beverley

If you want to use the BBC microcomputer as a dumb terminal to link up to a mainframe computer, you will probably have seen R.C. Rand's program in the July issue of *Acorn User*.

It was intended for use with the 0.1 operating system and there are a number of FX/OSBYTE calls which make it easier to write a dumb terminal program using the 1.0 system.

The program given below uses OSBYTE calls throughout and works when the Tube is fitted.

It uses the simplex mode – anything typed on the keyboard is not only sent down the RS423 but also echoed to the screen.

If you want it to work in the duplex mode where the device with which you are communicating echoes all the characters sent to it, simply delete lines 240 and 250.

A choice of OSWRCH or OSASCI at lines 180 and 250 determines whether a linefeed character is generated with each carriage return.

Program

```
 10 REM Dumb Terminal Program
 20 REM P.E.Beverley  6/8/82
 30 REM Only works on O.S. 1.0
 40 REM Works even if Tube fitted.
 50 OSASCI=&FFE3
 60 OSBYTE=&FFF4
 70 OSWRCH=&FFEE
 80 DIM CODE 50
 90 FOR J=0 TO 2 STEP2
100   P%=CODE
110   [OPT J
120   .RS423
130   LDA #&91
140   LDX #1
150   JSR OSBYTE\character in RS423 buffer?
160   BCS keyboard
170   TYA
180   JSR OSWRCH\or OSASCI for CRLF
190   .keyboard
200   LDA #&91
210   LDX #0
220   JSR OSBYTE\character in keyboard buffer?
230   BCS RS423
240   TYA
250   JSR OSWRCH\or OSASCI for CRLF
260   LDA #&8A
270   LDX #2
280   JSR OSBYTE\Put character in RS423 output buffer
290   JMP RS423
300   ]
310   NEXT
320 *FX 7,7
330 *FX 8,7
340 *FX 2,1
350 CLS
360 CALL CODE
```

OFF RECORDS...

The London ACORN
-BBC Centre Suppliers to schools and Colleges

OFFWARE **NEW!**
The new range of programs and routines backed by the OFF Records reputation!

ATILITY DISC for the ATOM
Contains no less than seven disc utility routines:

*Copy. . . . . (disc to disc)
*COPYT. . . . . . . (tape to disc)
*CopyD. . . . . . . (disc to tape)
*RENAME
*PURGE
*BACKUP
*AUTORUN

£25.00 p.p. $ VAT incl.

BBC GAMES with superb graphics:

FRUITWORM	(Model A)
FRUIT	(Model A)
JAMCAR	(Model A – 32K)
LABYRINTH	(Model A – 32K)

each **£9.00** p.p. & VAT inclusive

Atom:
Full hardware and software support

BBC:
Models A and B at £229 and £399.
Repair Service for all machines supplied by Acorn or OFF Records. Software now in stock.
Memory up-grades £26.99

Printers:
Seikosha 100: £215
Epson MX80FT/3: £385
Smith Corona Daisywheel (not a converted typewriter!): £485

Cassettes:
BBC-matched cassette recorders: £32

Monitors:
12" 18Mhz green screen monochrome: £110
14" colour £325
14" Sanyo PAL colour TV. Superb value: £235

Tapes:
Top Tape: See adverts in Radio Times.
OFF Records beats all published prices!

Stationery:
Continuous: Labels, Listing Paper, Word Processing Paper, Invoice/Statement.

Books:
Large selection of computer books and mags.

Also:
TORCH, NASCOM, GEMINI, GALAXY, QUANTUM, DRAGON, MICROPROFESSOR

Computer House, 58 Battersea Rise,
Clapham Junction, London SW11 1HH.
Telephone: 01-223 7730



ACORN PLUS

FREE! NEW 100 PAGE CATALOGUE FREE!

Control Universal stock Acorn, Rockwell, Cubit and fine peripherals. Send for our catalogue.

ACORN STOCK

Ring Control Universal on four telephone lines for technical advice and fast deliveries on all Acorn products – Eurocards, systems, Atoms, all software, networks components, connectors, spares – everything you need.

MORE: STOCK

Control Universal also stock Rockwell Computers, EPSON and TEC printers, BMC and MICROVITEC vdu displays, G.P.I. EPROM programmers and erasers, disks, stationery, memory and TTL chips.

CUBIT'
The CUBIT range is made by Control Universal and includes single board computers with 4K RAM and VIA i/o chip for 6502, 6802 and 6809 processors; CU-MEM memory card for eight 24 or 28 pin memory chips, with on board battery back up for CMOS RAM; CUBIO 64/80 channel digital i/o card; CUBAN eight bit analogue interface with 16 analog inputs, one analog output and 20 digital i/o channels; CU-KEY ascii keyboard.

'ATOM PLUS 17K RAM — £69 why 17k ?

– to fill in the gap from hex 3COO to 3FFF with 1K of static RAM, and provide 16k of dynamic RAM from 3OOO to to 7FFF. Uses 5v only devices, and fits in the standard Atom case.

Standard Eurocard size and bus connector.

'CU-DRAM' 64K bytes DRAM — £129

For all Acorn and Control Universal systems. Each block of 4k can be enabled or disabled to match the system. Carries also a 4k/8k 28 pin socket for ROM or EPROM, and can be software selected at board level to allow up to 16 boards in one system and hence a maximum of 1Mbyte of RAM.

Standard Eurocard size and bus connector.

CONTROL UNIVERSAL LTD.
Unit 2, Andersons Court, Newnham Road, Cambridge
(0223) 358757

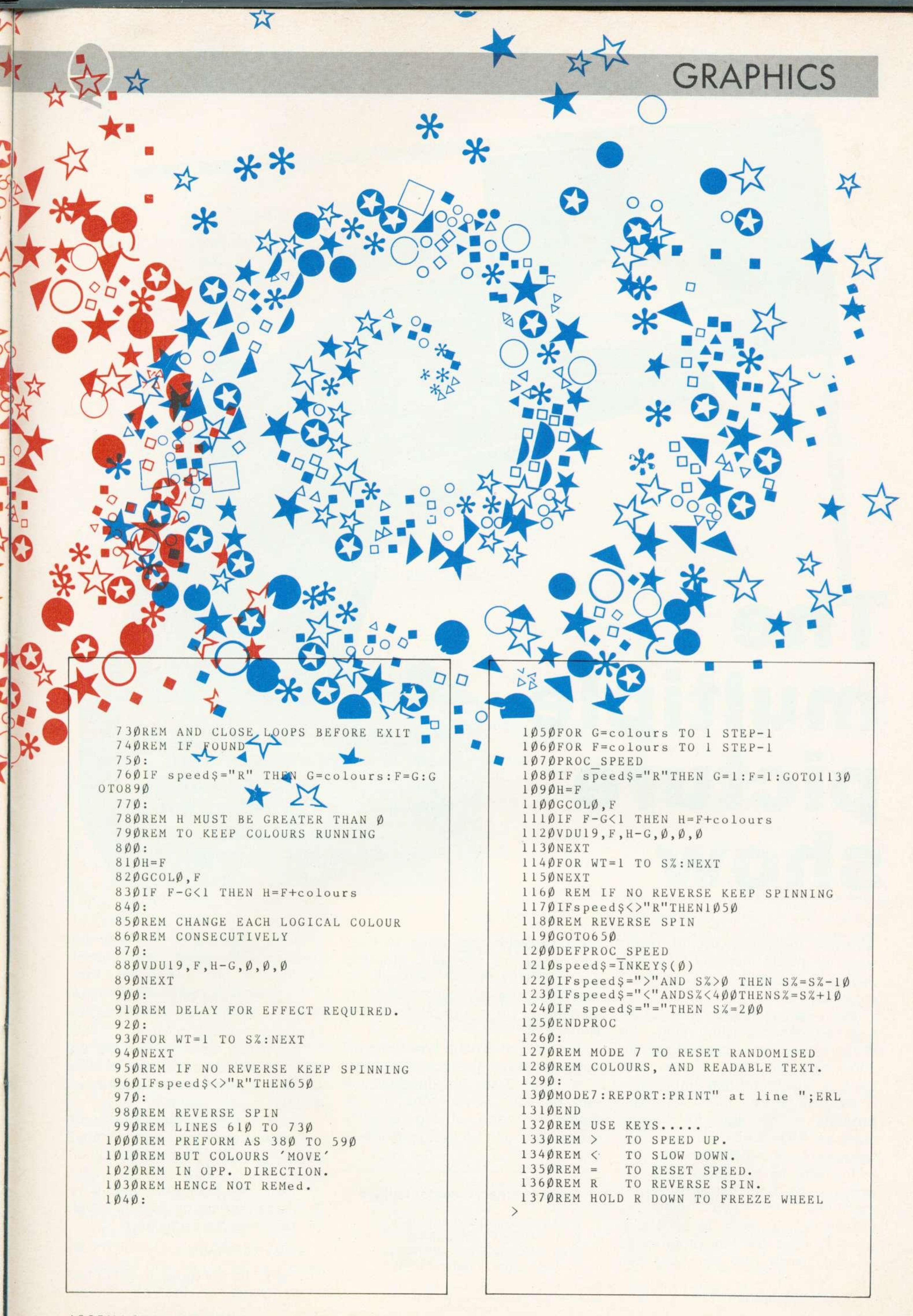
VISIT OUR NEW SHOWROOM

FIREWORKS

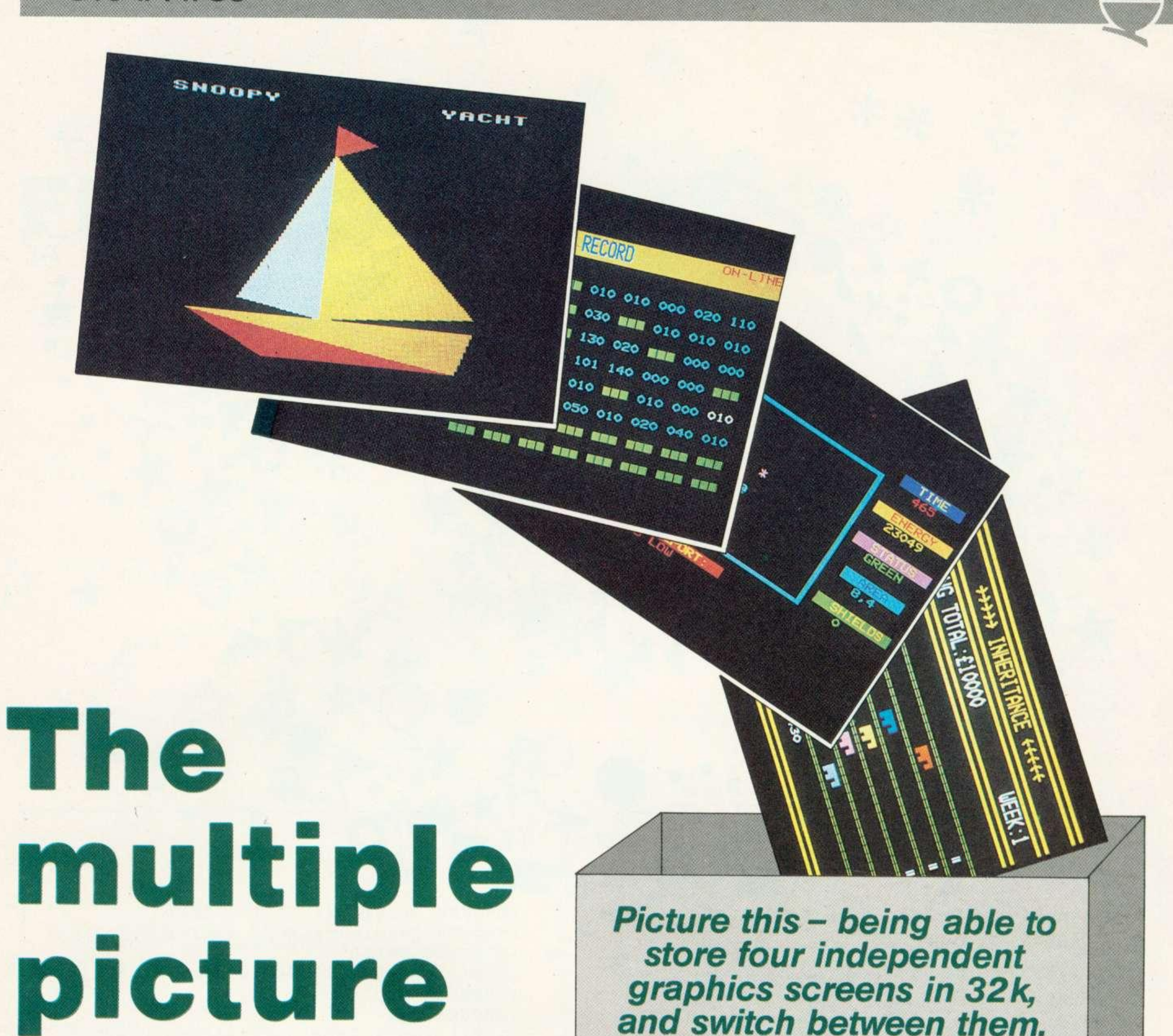



```
 LOAD"8SPRL"
>L.
  1ØØREM CATHERINE WHEEL
  11ØREM J.TELFORD.
  12Ø:
  13ØON ERROR GOTO13ØØ
  14ØMODE7
  15Ømode=5
  16Øcolours=3
  17Ø:
  18ØREM CHECK FOR MODEL A
  19Ø:
  2ØØIF HIMEM = 31744 THEN mode=2:colou
rs=7
  21Ø:
  22ØREM USE BEST MODE+COLOURS
  23Ø:
  24ØMODE mode
  25Ø:
  26ØREM SET UP SPEED,RADIUS,
  27ØREM CURVE FACTOR AND
  28ØREM INITIAL ANGLE OF TURN
  29Ø:
  3ØØS%=2ØØ
  31ØR=896
  32ØCF=1Ø
  33Øturn=Ø
  34Ø:
  35ØREM TURN OFF CURSOR
  36Ø:
  37ØVDU23;82Ø2;Ø;Ø;Ø;
  38Ø:
  39ØREMEMBER TO CENTRE THE ORIGIN
  4ØØREM ON THE SCREEN.
```

```
  41Ø:
  42ØVDU29,64Ø;512;
  43Ø:
  44ØREM MOVE TO START OF CURVE
  45ØREM WITHOUT UGLY LINES
  46Ø:
  47ØMOVE R*COS turn,R*SIN turn
  48Ø:
  49ØREM WE NEED 351 SEGMENTS
  5ØØ:
  51ØFOR C%= Ø TO 351
  52Ø:
  53ØREM CHANGE COLS FOR EACH SEGMENT
  54Ø:
  55ØGCOLØ,C% MOD colours+1
  56ØDRAWR*COS turn,R*SIN turn
  57ØDRAW(R-5Ø)*COSturn,(R-5Ø)*SINturn
  58Øturn=turn+PI/CF:R=R-CF/4
  59ØPLOT85,R*COS turn,R*SIN turn
  6ØØPLOT85,(R-5Ø)*COS turn,(R-5Ø)*SIN
turn
  61ØNEXT:REM SPIRAL DONE NOW
  62ØREM PRESS ANY KEY TO SPIN WHEEL
  63Ø:
  64Øa$=GET$
  65ØFOR G= 1 TO colours
  66ØFOR F=1 TO colours
  67Ø:
  68ØREM CHECK FOR SPEED UP OR DOWN
  69Ø:
  7ØØPROC_SPEED
  71Ø:
  72ØREM CHECK FOR REVERSE DIRECTION
```



```
 730REM AND CLOSE LOOPS BEFORE EXIT
 740REM IF FOUND
 750:
 760IF speed$="R" THEN G=colours:F=G:G
OTO890
 770:
 780REM H MUST BE GREATER THAN 0
 790REM TO KEEP COLOURS RUNNING
 800:
 810H=F
 820GCOL0,F
 830IF F-G<1 THEN H=F+colours
 840:
 850REM CHANGE EACH LOGICAL COLOUR
 860REM CONSECUTIVELY
 870:
 880VDU19,F,H-G,0,0,0
 890NEXT
 900:
 910REM DELAY FOR EFFECT REQUIRED.
 920:
 930FOR WT=1 TO S%:NEXT
 940NEXT
 950REM IF NO REVERSE KEEP SPINNING
 960IFspeed$<>"R"THEN650
 970:
 980REM REVERSE SPIN
 990REM LINES 610 TO 730
1000REM PREFORM AS 380 TO 590
1010REM BUT COLOURS 'MOVE'
1020REM IN OPP. DIRECTION.
1030REM HENCE NOT REMed.
1040:
1050FOR G=colours TO 1 STEP-1
1060FOR F=colours TO 1 STEP-1
1070PROC_SPEED
1080IF speed$="R"THEN G=1:F=1:GOTO1130
1090H=F
1100GCOL0,F
1110IF F-G<1 THEN H=F+colours
1120VDU19,F,H-G,0,0,0
1130NEXT
1140FOR WT=1 TO S%:NEXT
1150NEXT
1160 REM IF NO REVERSE KEEP SPINNING
1170IFspeed$<>"R"THEN1050
1180REM REVERSE SPIN
1190GOTO650
1200DEFPROC_SPEED
1210speed$=INKEY$(0)
1220IFspeed$=">"AND S%>0 THEN S%=S%-10
1230IFspeed$="<"ANDS%<400THENS%=S%+10
1240IF speed$="="THEN S%=200
1250ENDPROC
1260:
1270REM MODE 7 TO RESET RANDOMISED
1280REM COLOURS, AND READABLE TEXT.
1290:
1300MODE7:REPORT:PRINT" at line ";ERL
1310END
1320REM USE KEYS.....
1330REM >   TO SPEED UP.
1340REM <   TO SLOW DOWN.
1350REM =   TO RESET SPEED.
1360REM R   TO REVERSE SPIN.
1370REM HOLD R DOWN TO FREEZE WHEEL
>
```



The multiple picture show

Picture this – being able to store four independent graphics screens in 32k, and switch between them. Peter Goode explains the principle

Imagine the advantage of being ables to switch instantly from one picture to another on your BBC microcomputer.

Many games require a complex, high-resolution playing field, for instance, and drawing one takes an annoyingly long time – even with the BBC machine's fast graphics.

How pleasant, then, for your program to be able to draw a complex maze while you are busy reading – or understanding the instructions for the game. And it is possible, on both model A and B machines, for two effectively independent screens to exist side by side, and for the program to choose which one the user can see.

No incomprehensible machine code or peek and poke routines are necessary. Standard graphics commands can be adapted to provide complete control over two screens, or up to four on the model B machine.

First let me explain how the BBC computer stores graphics information.

In mode 5, the only mode which can provide two screens on the model A, each of the screen's 40960 dots can be assigned one of four colours.

Table 1. Binary code for colours

A	B	
0	0	= Colour 0, default black
0	1	= Colour 1, default red
1	0	= Colour 2, default yellow
1	1	= Colour 3, default white

Two binary bits per dot are needed to store this much information.

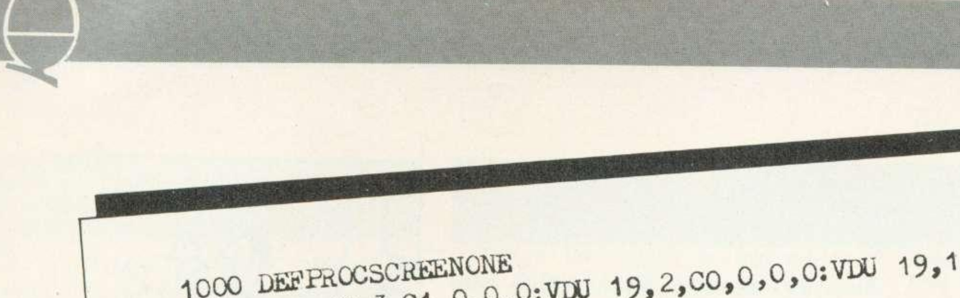
Each of these bits can be either 0 or 1, and only two are necessary to give one of four colours.

The binary numbers to represent the four colours are shown in table 1. The two bits have been assigned arbitrary labels, A and B.

If we choose colour 1 as 'screen 1 foreground colour' and colour 2 as 'screen 2 foreground colour', ignoring bit A allows us to draw in colour 1 on a background of colour 0. We can ignore bit A quite simply by executing the command:

```
VDU 19,2,0,0,0,0
```

This tells the computer to set colour 2 to black, ie the background




```
1000 DEFPROCSCREENONE
1010 VDU 19,3,C1,0,0,0:VDU 19,2,C0,0,0,0:VDU 19,1,C1,0,0,0
1020 S=1:PROCPLOT
1030 ENDPROC
1040 DEFPROCSCREENTWO
1050 VDU 19,3,C2,0,0,0:VDU 19,1,C0,0,0,0:VDU 19,2,C2,0,0,0
1060 S=2:PROCPLOT
1070 ENDPROC
1080 DEFPROCPLOT
1090 GCOL 1,S
1100 ENDPROC
1110 DEFPROCERASE
1120 GCOL2,3-S
1130 ENDPROC
1140 DEFPROCINIT
1190 MODE 5
1200 C0=0:C1=7:C2=3
1210 PROCSCREENONE:PROCPLOT
1220 ENDPROC
```

colour, thus making all lines previously drawn in colour 2 invisible. The command:

```
VDU 19,1,0,0,0,0
```

followed by

```
VDU 19,2,3,0,0,0
```

will blank out lines drawn in colour 1 and restore those drawn in colour 2 to their original colour (yellow). Colour 2, of course, can be set to any colour desired.

To go into further detail would probably confuse. Nothing beats hands-on experience with computers so here are some working routines and examples. All require the procedures shown in program 1, so type them in before advancing any further.

The procedures perform the following tasks:

- PROCSCREENONE will display screen one and enable plotting on screen one.
- PROCSCREENTWO will perform the same task but for screen two.
- PROCPLOT will turn on plotting on the screen selected by S.
- PROCERASE will turn on erasing on the screen selected by S.
- PROCINIT switches to mode 5, screen one and enables plotting on that screen.

Try the following:

```
PROCINIT
```

The computer will switch to mode 5 graphics and text. You can now draw on screen one. Try entering:

```
MOVE 0,0
MOVE 1000,1000
PLOT 85,1000,0
```

This will cause the computer to draw and fill a triangle, in white.

```
PROCSCREENTWO
```

'No machine code or peek and poke routine necessary'

The triangle has vanished.

```
PROCPLOT
MOVE 0,0
MOVE 0,1000
PLOT 85,1000,0
```

A new triangle has appeared.

```
PROCSCREENONE
```

the old one appears again,

```
PROCSCREENTWO
```

the new one again.

The background and two foreground colours can be set to any colour (0–15) at any time by changing the values of C0, C1 and C2. For example, try:

```
C0= 1
C1 = 11
PROCSCREENONE
```

The first triangle re-appears but in flashing yellow and blue on a red background.

The screen one and screen two procedures work well for this, but if we want to draw on the invisible screen, and not the one the user can see, we simply use

```
S = 1
PROCPLOT
```

to draw on screen one, and

```
S = 2
PROCPLOT
```

to draw on screen two.

Erasing can also be done in this way using PROCERASE instead of PROCPLOT. For example:

```
PROCSCREENONE: S=2: PROCPLOT
MOVE 0,0:MOVE 1000,1000:
PLOT 85,0,1000
PROCSCREENTWO:REM make it visible.
```

If you are lucky enough to have 32k of RAM try adapting the procedures to create two screens of four colours or four of two.



Beeb link to second micro

Sir, Is it possible to link my ITT computer up to a BBC micro via the RS232 interface, and will the Tube or 1MHz extension bus allow this?

I am also using a disc and printer with my ITT system.

Mr J. Hughes
Liverpool

The BBC micro has an RS423 interface which is compatible with RS232C equipment, and will transmit over greater distances. The RS423 allows you to transmit or receive serial data, ie, to act as a dumb terminal with a larger computer transferring serial ASCII data. It also allows you to use a serial RS232C type printer.

The information the micro gives is as follows:

1 START BIT
8 BITS OF DATA
1 STOP BIT
NO PARITY

Some printers require either even or odd parity, therefore they will not work with the micro, however, there may be internal links which will give the printer the above format.

With the 0.1 machine operating system the micro will not receive serial data from another micro, using the simple *FX commands, but it will transmit. (See July *Acorn User* for which will allow the user to receive this information).

The version 1.0 will allow you to transmit and receive using the single FX commands which are available.

Serial data will consist of ASCII code which can be placed in either machine's RAM, therefore programs will only run on one of the machines because the basic and memory maps will be different.

The Tube will be a high speed data link from a second processor to the BBC micro. It probably won't allow you to connect another computer to this, but fuller details will be available shortly.

The 1MHz extension bus will allow you to connect Acorn's eurocards to the micro. It will then be possible to add extra RAM cards to the micro without having to purchase a second processor.

Atom packs up on Basic

Sir, About 18 months ago I purchased an Atom micro, which after about three months of disuse refused to enter Basic on power up.

The display was a seemingly random collection of characters, and repeated depression of break did not produce a reset, neither did powering down and up repeatedly.

Examination of the break circuit shows no abnormality, and the PSU shows no unusual fluctuations. I have not tried directly manipulating the 650Z reset line (mainly as I'm not sure which pin carries it!)

Mr S. Calsy
Herts

Your machine will have to be returned to Acorn's service department, as with problems like this quite a bit of circuitry is used.

The actual break key could be at fault, the 8255 I/O chip could be at fault, ie, not accepting input from break key, or giving an incorrect clock frequency for 6502 (1MHz). If readers have any problems with their Acorn micros, the first port of call is the local dealer (see page 72).

Retail control systems take care of faults dealers cannot handle. The address is Gresham House, Twickenham, Road, Feltham, Middlesex TW13 6HA. If you need advice telephone TCS on 01-898 4761.



User group news – page 48

Computer theft in the night

Sir, Could you find room to tell readers that my BBC model B microcomputer, serial number ICL 006847 has been stolen. It disappeared during the night along with other goods when thieves broke into my home.

If anyone knows of its whereabouts, would they contact their local police, or let me know through the magazine.

To lose the computer when I had only had it four weeks is most upsetting. I was just beginning to enjoy it.

Roger Bishop
Kent

String errors

Sir, I have a program that inputs 10 series of 10 strings, 10 integer variables and ten floating point variables.

However, when the computer reads the data, it misses something in the sixth series, seventh string and gives an error message.

String number seven then contains peculiar things. How can I stop this happening?

Mogens Johannsen
Denmark

Entering the following few lines at the beginning of your program should make it work.

```
5 !&70 = &F5212048
6 !&74 = $6068
7 ?¾218 = 0
8 ?&219 = &70
```

Hot stuff

Sir, I have a model B BBC micro which gets hot from the power supply. Will this affect my micro, and will the power supply allow for the addition of a disc unit?

D. Hill
Birmingham

Sir, After running my BBC model B, which I have had for about a week, for several hours it began to give off the smell of warmed plastic. These fumes were obviously caused by the heat of the power supply unit, and the computer's ventilation holes were not obstructed in any way.

What I would like to know is whether this is just a temporary problem because of the newness of the machine, or if not, what can be done.

Mr D. Short
Cheshire

The linear power supply by its very nature gets hot, but this should not effect the running of your machine.

If it gives off fumes, there must be something wrong with the machine and it should be returned to your nearest dealer for checking.

When the micro is upgraded to take a disc interface at an approved service centre, the power supply unit is changed free of charge to a switched mode unit at the same time.

Pascal ROM

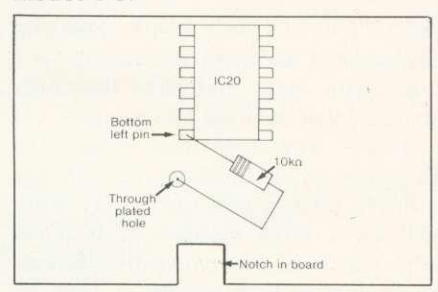
Sir, Is the Pascal ROM available yet? How much is the ROM, which dialect will it be?

Certain dealers have offered a model A with a 32k RAM. Am I right in believing that this computer will carry out any of the graphics and any other capabilities (other than the bus etc)? Would this run Pascal?

Mr C. Smith
Kent

We have no details on Pascal at present, but Acorn will be releasing a version probably in ROM. The exact date is not known, but it should be in spring next year. The cost has yet to be fixed.

With the additional RAM you will be able to obtain the extra graphics modes 0-3.



Buzz off

Readers who are fed up with a buzzing BBC micro can solve the irritation by connecting a 10kΩ resistor to IC20 as shown above. This chip is found on the main board near the front of the micro under the keyboard.

This alteration is only recommended for people who are used to working with electronics, as it will invalidate Acorn's guarantee. Dealers can also make this change.



Teachers with an urge to program

Sir, First of all, thank you for September's excellent *Acorn User* – I hope you can maintain the high standard you have set yourselves.

I am convinced that when the Department of Industry scheme for primary schools gets under way there will be a large number of teachers who will wish to involve themselves, in no matter how modest a way, in programming on their newly-acquired computers, rather than rely solely on commercial software.

As a primary school teacher with a BBC micro, I am interested in contacting other teachers (or parents) who would like to develop and exchange short, simple programs (simple from a programming point of view, that is) suitable for children aged 5 – 11. These would complement the more complex programs that will become available.

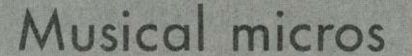
In addition, there could be an exchange of teaching suggestions, sample worksheets etc. for these programs, and for other, professionally-produced programs.

How formal and structured a group we might become would obviously depend on numbers interested and their views.

J.A. Sheard
31 Glen Court
Avenue Rd
Wolverhampton
WV3 9JN

It is the policy of *Acorn User* to publish such programs, and the Editor will give whatever support he can to encourage communication between teachers.





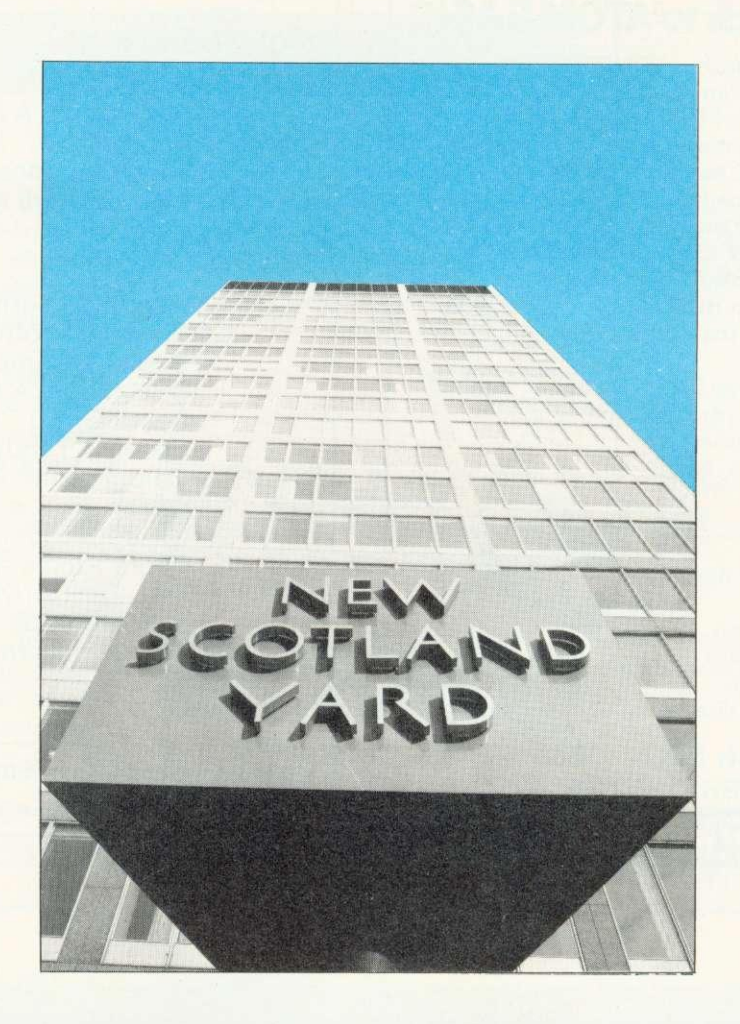
Musical micros

Sir, I am writing to ask whether BBC micros can be connected to conventional synthesisers to trigger sequences or store note and chord progressions. Most synthesisers now have sockets to trigger drum machines etc, and 'programmable' synthesisers with tape storage capability are astronomically expensive.

Brett Jordan
Middx.

Can anybody out there help? Mr Jordan added a p.s. to his letter. Apparently video game addicts in the US are known as 'vidiots'. We like it!





ELECTRONIC ARMS

Simon Dally experiences first-hand how computers help the long arm of the law reach further and faster to beat crime

Last month we looked at the growth area of computer crime. But the computer also has proved more than a match for many criminals and is now in the forefront of today's fight against crime.

The most common felonies involve motor vehicles: stolen cars, drunken driving, speeding, bald tyres and so on. Most vehicle information is stored on the infamous Swansea computer, run by the Department of the Environment.

In spite of the widespread criticism it attracted when first installed in the early 1970s – it cost several times the original estimate and the delays were appalling – it now seems to function very efficiently at least in one respect as I recently discovered.

One night as I was driving over Battersea Bridge I passed a police car and was waved down by a second on the other side. I was astonished to be greeted by the police officer in his 'just a routine check, sir' voice by my own name!

In the course of the 30 seconds it had taken to cross the bridge, my car number had been radioed to the computer and the details had been radioed back! Well, that's my story and I'm sticking to it . . .

From the police point of view, the most impressive technological advance in recent years has been the computerisation of the National Fingerprint Collection at New Scotland Yard.

Six years ago the system was entirely manual. The collection comprised more than 2½ million prints, being added to at the rate of 200,000 a year and dealing with about 2000 inquiries a day from around the country.

Since the average inquiry entailed making comparisons with 50 other prints – all stored on paper – the clerical effort involved was immense. Moreover, the prints could be consulted only by one officer at a time and the possibility of filing errors was all too likely.

The system the police have now adopted is Videofile and comes

page 70 ▶



MC 370M R.G.B. COLOUR MONITOR

Cabel offers you a high quality 14 inch colour monitor

This monitor will clearly display 80 x 25 characters, unlike most of our competitors'. The MC 370M has been designed, using the highest quality materials and the most sophisticated components available on the market today, ensuring long service and reliability with high technology.

Our "2 Year Guarantee" includes parts and labour, proving our faith in our products, which have been widely used on the manufacturing markets for the past 10 years.

Price £289.00 including VAT and delivery.
Special offer this month, Free R.G.B. Lead.

CHEQUE / POSTAL ORDER, OR FOR FAST DELIVERY, RING US WITH YOUR CREDIT CARD NUMBER



Mount Road, Burntwood, England, WS7 0AX
Telephone: 021-308 7075 Telex: 339671



▶ *from page 67*

from California. In essence it combines video techniques with sophisticated computer process control.

The computer stores the information in a digital database and keeps track of where each set of prints is on tape. The visual display uses a split-screen which magnifies the picture many times, and provides a resolution more than twice that of a standard TV screen. This enables suspects' prints to be shown simultaneously with the possible matching sets, enabling comparisons to be made quickly and by several officers at once.

The computerisation of fingerprints and criminal records has clearly saved a vast amount in terms of manpower and physical space. It is, however, no panacea – as the mounting crime figures and relatively low detection rates testify.

'Computerisation is, however no panacea'

But the police are understandably reticent when it comes to discussing their use of computers. Every few months a story seems to break in the popular press of a breach of security in their system which gives an unauthorized person access to confidential information. The press likes to imagine sinister computers 'chatting' to each other.

On the other hand, the press put up a tremendous howl about police incompetence during the hunt for the Yorkshire Ripper – and after his capture. But while the police have since conceded the investigations did become bogged down in paperwork, they point out that they tried to use the police national computer and found it too difficult to put all the information on to it.

And there are the real and more worrying fears of what the security services and Special Branch get up to with their computers. This familiar and very serious problem will not be easily resolved.

SEPTEMBER QUIZ RESULTS

We received over 150 correct entries for our codebreaking competition in the September issue The solutions, for those who tried and failed, are printed on the right.

In the second example, the keyword was 'JANE AUSTEN' and the text was from 1 *Corinthians* XIII.

We apologise for the widespread gremlin who slipped in a misprint and inserted two wrong spaces – this didn't seem to put any of you off and in the real world coding operators make errors too! Our thanks to Brian Gill of Oldham for a most entertaining discourse on the nature of *Gremlinus Substituens*.

The winners, chosen at random, were

Anna Rose Surguy of Reading, Berks
Dave Woolcock of Preston, Lancs
Peter Gordon of Currie, Lothian

They all win £20-worth of programs for the BBC micro from Acornsoft.

To those snivelling wretches who complained about the competition being too easy, please bear in mind that we have to try to strike a balance for our readers, many of whom are totally new to microcomputers and have relatively modest mathematical abilities: we aren't trying to run the house journal of Mensa.

However, please stay with us: we are planning something which should silence all your carping noises for a while!

WEATHER REPORT FROM THE TIMES JULY FOURTEEN NINETEEN EIGHTY TWO LONDON SOUTH EAST ENGLAND THUNDERY SHOWERS BECOMING MORE WIDESPRIVE WIND EASTERLY LIGHT TO MODERATE TEMPERATURE TWENTY F IVE DEGREES CENTIGRADE SEVENTY SEVEN FAHRENHEIT

WHEN I WAS A CHILD I SPAKE AS A CHILD I UNDERSTOOD AS A CHILD I THOUGHT AS A CHILD BUT WHEN I BECAME A MAN I PUT AWAY CHILDISH T HINGS FOR NOW WE SEE THROUGH A GLASS DARKLY.

CODEWORD – "JANE AUSTEN"



WIN A BBC MICRO

A BBC microcomputer worth £299 is the prize in this month's competition set by Simon Dally

A puzzle editor once set a problem as follows: find a nine-digit-number which contains all digits from one to nine inclusive. The full number is exactly divisible by nine; knock off the right-hand digit and the number is exactly divisible by eight; knock off another digit and the result is exactly divisible by seven, etc – until you are left with one digit which is, of course, exactly divisible by one.

Pleased with his research, our editor declared that there was only one number which satisfied these conditions. But to his horror, a stream of abuse landed on his desk claiming there were several solutions. Your task is to check the puzzle out. Was the editor right, in which case what is the solution? Were the readers right, in which case how many solutions are there?

Entries should be sent to the Competition Editor, Acorn User, 53 Bedford Square, London WC1B 3DZ, to arrive by December 1



Official Acorn Dealer List in the UK

Acorn dealers stock and service the Atom computer, Acorn systems and Acornsoft software. Many also offer service facilities for the BBC micro and these are marked with an asterisk. The Acorn dealer not only sells computers and peripherals but provides vital customer support. Most have recently attended technical seminars in Cambridge to ensure that they deal effectively with customers' enquiries. In addition, Acorn supply dealers with specific test and diagnostic equipment to speed fault finding.

LONDON
Cavendish Sales E1
01-247 3453
Canonbury Radio
01-226 9392
*Centre of Soundwill
01-727 0511
*Group 70 E18
01-505 7724
*Micro Store SW3
01-352 9291
*Off Records SW11
01-223 7730
Paul Electrical SW20
01-542 6546
PAJ Rental Ltd NW4
01-368 0572
REW West End Video Centre WC2
01-240 3386/7
RA Bailey Ltd SE26
01-693 1818
*Technomatic Ltd NW10
01-723 0233
The Video Palace W8
01-937 8587

BIRMINGHAM
Quality Radio & Television Co B14
021-444 2889
Typewriter Centre (Erdington) B23
021-382 0185
Typewriter Centre (Head Office) B5
021-622 5385/7
Typewriter Centre (Dudley Road) B18
021-455 9111
Typewriter Centre (Kings Heath) B14
021-444 7349

AVON
*Microstyle Bath
0225-334659

BEDFORDSHIRE
*Broadway Electronics Ltd Bedford
0234-213639

BERKSHIRE
*Windsor Computer Centre Windsor
07535-58077

BUCKINGHAMSHIRE
A L Wheeler Ltd Great Missenden
024-06 2560
*Tarace Ltd Wendover
0296-623915
Hi-Vu Electronics Wolverton
0908 312808

CAMBRIDGESHIRE
*Arden Personal Computers Peterborough
0733-47767
*Cambridge Computer Store Cambridge
0223-65334
*Control Universal Cambridge
0223-358757

CHESHIRE
Bellard Electronics Ltd Upton
0244-380123
*Northern Computers Warrington
0928-35110

CLEVELAND
*Customised Electronics Ltd Middlesbrough
0642-247727

CORNWALL
*Brewer & Bunney Camborne
0209 712681
Microtest Ltd Bodmin
0208 3171

DERBYSHIRE
*Datron Micro Centre Derby
0332-380085
*First Byte Computer Systems Ltd Derby
96 365280

DEVON
*Bits & Bytes Ilfracombe
0271-62801
Devon Computers Paignton
0803-526303
J.A.D. Integrated Services Plymouth
0752 62616

DORSET
Landsowne Computer Centre Bournemouth
0202 23776

ESSEX
*Arden Business Systems Colchester
020-637 368
*Computers for All Romford
0708-60725
Emprise Ltd Chelmsford
0245 356834
Emprise Ltd Clacton on Sea
0255 431296
*Emprise Ltd Colchester
0206 865926

GLOUCESTERSHIRE
Computer Shack Ltd Cheltenham
0242 584343
Independent Computer Consultants Tewkesbury
0684 298033

HAMPSHIRE
*Electronequip Porchester
0705 325 354
Ferranti & Craig Ltd Basingstoke
0256 52203
Hugh Symons Group of Companies Bournemouth
0202-23776
*R D S Electrical Portsmouth
0705-812478
*R M K Electronics Ltd New Milton
0425-616110

HEREFORDSHIRE
Kempson's Hereford
0432 3480

HERTFORDSHIRE
*Compshop Ltd New Barnet
01-441 2922
*Computer Plus Watford
0923 33927
H F Sheffield Ltd Abots Langley
092 77 63184
Intelligent Artifacts Ltd Royston
0233 207689
*Q-Tek Systems Ltd Stevenage
0438 65385

HUMBERSIDE
*Computer Facilities Scunthorpe
0724-63167
*Vixon Computer Systems Cleethorpes
0472 58561

KENT
Kent Computers Herne Bay
02273-68900
*M D Wright Data Services Canterbury
0227-69090

LANCASHIRE
Almaine Co Colne
0282 863520
J Lambert (Radio) Ltd Burnley
0282 71459
Merit Computers Ltd Wigan
0942-495821
Microrose Ltd Birmingham
0468 62180
*NSC Computing Shops Ltd Manchester
061-832 2269

LEICESTERSHIRE
*D A Computers Leicester
0533 549407
Percy Lord & Son Ltd Wigston
0533 785033
Pratt Bros (Leicester) Ltd Blaby
0533 773020
*R H Murcott Ltd Loughborough
0509 214444
*TB Computer Services Market Harborough
0858 32501

LINCOLNSHIRE
*Oakleaf Computers Grantham

LIVERPOOL/MERSEYSIDE
BEC Computer World Liverpool
*Data Exchange Ltd Birkenhead
051-647 4213
*Liverpool Computer Centre Liverpool
051-236 2000
Tisdalls Newton-le-Willows
09252 5577

MIDDLESEX
*Microage Electronics Edgware
01-959 7119
*Twickenham Computer Centre Twickenham
01-891 1612

NORFOLK
*Anglia Computer Store Norwich
0603 29651

NORTHAMPTONSHIRE
MA Electrical Installations Irthlingboroug
0933 650133

NORTHUMBERLAND
Tom Newton & Sons Wooler
066 82221

NOTTINGHAMSHIRE
*Leasalink Viewdata Ltd Nottingham
0602 396976
H N & L Fisher (Huthwaite) Ltd Sutton in Ashfield
0623 53435
*Mansfield Computers & Electronics Mansfield
P R Hartley Ruddington
0602 213492
S P Electronics Hucknall
0602 632467

SHROPSHIRE
Com-tel Newport
*Jentech Services Ltd Bridgenorth
07462 5287

SOMERSET
*Somerset Business Computers Taunton
0823-52149
*The Computer Room Yeovil
0935-20268

STAFFORDSHIRE
Typewriter Centre Wolverhampton
90-27627

SUFFOLK
CE Matthews & Co Ipswich
0473 215666
Emprise Sudbury
0787 310110
*Micro Management Ipswich
0473 59181
Midwich Computer Co. Ltd Rickinghall
0379 898751
S Emery & Co Bungay
0271 62503

SURREY
*JS Simnett Computers Ltd Kingston upon Thames
01-546 3793
*PJ Equipments Ltd Guildford
0483-504801
*3D Computers Surbiton
01-337 4317

SUSSEX
*Castle Electronics Hastings
0424-437875
*Datex Micros Worthing
0903-39290
Gamer Brighton
0273-698424
*Microcentre Bognor Regis
0243 827779

TYNE AND WEAR
* H C C S Gateshead
0632-821924
Newcastle Computer Services Newcastle-upon-Tyne
0632-761168

WARWICKSHIRE
Carvell Rugby
0788 65275

WEST MIDLANDS
A E Chapman & Co (Old Hill) Ltd Cradley Heath
D F Gibbs Ltd Coventry
0203 87432
H & H Business Systems West Bromwich
021-553 7606/7
Ibek Systems Coventry
0203-661162
Micrologic Ltd Halesowen
021-550 8036
Richard Morris (Electrical) Ltd Warley
021-429 1161
*Taylor Wilson Systems Ltd Solihull
05645 6192
Typewriter Centre Coventry
0203-28799
Typewriter Centre Walsall
92-27589
Typewriter Centre Sutton Coldfield
021-355 6789
*Walters Computer Systems Ltd Stourbrid
03843-70811/2/3

YORKSHIRE
Arthur Yates Ltd Ripon
0765-2737
*Customised Electronics Ltd Leeds
0532-792332
*Datron Computers & Supplies Sheffield S10
0742-755105
*Datron Micro Centre Sheffield S7
0742-585490
*Eltec Services Ltd Brdford
0274-491371
Green's Telecom Barnsley
0226-5031
Micro Power Leeds LS7
0532-683186
*Superior Systems Ltd Sheffield 1
0742-755005

WALES
*Bucon Ltd Swansea
*Cardiff Microcomputers

SCOTLAND
*Esco Computing Glasgow
041-4275497
*Highland Microcomputers Inverness
0463-221544
Ian Hutchison (Leven) Ltd Leven
0333-23870
J H Donald Ltd Hurlford
McQueen Systems Ltd Galashiels
0896W M Coupar Blairgowrie & Dundee
0382 2436/7

NORTHERN IRELAND
*C E M Microcomputer Services Belfast
0232-44111/43564

CHANNEL ISLANDS
*Channel Data Processing (Jersey) St Helier
0534-24333/72897

ISLE OF MAN
*Typestyle Ltd Onchan
0624-25890/24650

ISLE OF WIGHT
Excell Bembridge
098 387 2578



Function key programs

```
 10 *KEY0MODE6|MVDU19;4;0;|M|N|ML.|M
 20 *KEY1RUN|M
 30 *KEY2LIST07|M
 40 *KEY3LIST00|M
 50 *KEY4$&DF0="
 60 *KEY5SAVE$&DF0|M
 70 *KEY6LOAD""|M
 80 *KEY7CHAIN""|M
 90 *KEY8VDU12,23;12;0;0;0;|M
100 *KEY9AUTO
110 *KEY10OLD|M
120 INPUT N$
130 IF N$="" THEN N$="PROG"
140 $&DF0=N$
```

Ten ideas for function keys

When developing programs on the BBC microcomputer it is helpful to have the function keys programmed to give frequently used commands. Everyone has their own favourites, such as RUN, LIST07 LISTOO, LOAD"", CHAIN"", AUTO, and for the break key, OLD – but some are not.

Above key 0 on my computer it just says LIST, but this is actually composed of a number of commands. Select mode 6, change the background colour to blue – giving the effect of individual lines being separated by black lines as seen on BBC's *Computer Programme* – set paged mode (CTRL N), add an extra carriage return and then list the program.

The extra carriage return gives a total of three lines above the first line of the program so if you want to look at the first page of a long program and press ESCAPE, you do not lose the first line off the top of the screen.

Some explanation is also needed for keys 4 and 5, and along with those you will need to refer to lines 120 to 140 of the setting-up program. When you want to SAVE a program you need to specify a filename (unlike LOAD and CHAIN) which makes using a function key a little awkward.

My first idea was to put SAVE" on to a function key and then add the name and close the quotation marks from the keyboard. I then tried having SAVE N$M on a key and declaring the name by typing N$ = "FRED" or whatever. The problem here is that N$ is a dynamic variable and disappears as soon as you play around with the program.

I finally hit on the idea of storing the name in a free chunk of memory in the workspace below the program and accessing it by using string indirection.

So from the initialisation program or from the keyboard you can set up the filename and leave it until you move on to another program.

Thus to save a program twice and then rewind the tape consists of: press key 5, start the recorder, press RETURN, key 5, RETURN, and then do a "*. RETURN" (an abbreviation for *CAT).

Having rewound the tape it is possible to do a catalogue to verify the program, and when finally rewound, press ESCAPE to turn off the motor.

This explains the lack of *MOTOR1 and *MOTOR0 from the function keys – it is so easy to type in "*.RETURN" and then press ESCAPE, that it is not worth using up function keys on it.

But a word of warning. Since I have chosen &DFO which is just below the program at &E00, if you try to use a filename of more than 15 letters you will overwrite the beginning and get a "Bad Program" prompt!

The only other key that needs any explanation is key 8. This clears the screen (VDU12) and then you can use the advanced graphics call VDU 23, 0 to tell the VDU driver chip, which is programmable, that video RAM starts at page zero.

This has the effect of displaying a 'bit-map' of zero page, all the workspace and some of the program area depending on which mode of graphics you are in. But it only works effectively in a two-colour mode (not mode 7) and is most easily visible with white on a blue background.

It can be set up by pressing key 0, ESCAPE (if it's a long program to be listed) and then key 8. This view of the computer workings is not only fascinating ut also a useful diagnostic aid.

Paul Searle

EDITING FROM TOP TO BOTTOM

Editing programs on the BBC microcomputer is easier if you remember that when the text cursor is at the bottom of the screen and you want to edit a line in the upper half, it is quicker to move the cursor down rather than up.

As it goes off the bottom of the screen it reappears at the top. Indeed if you have three or four consecutive lines to edit it is worth pressing the RETURN key until the first line to be edited is at the top of the screen. To reach it then only requires a single press on the cursor down key.

Having finished editing, you press RETURN, and this brings the next line to the top of the screen and again only needs a single cursor down to reach it.



FIGHTING BACK

Struggling with a family of computer maniacs is no joke, but **Mrs Ronnie Rowsell** finally gets some joy

The day after the micro had arrived, a man came to repair the television. The rest of the family were greatly cheered, but I wasn't bothered, because the mono set was the only one I had access to.

A few days later the tribe organised some sort of rota system for using 'Brains' (the chosen name). The system and the machine seemed to both work smoothly. I was secretly becoming interested in joining in – my curiosity was getting the better of me. But I couldn't use the thing because I didn't know how to and it would take too long for a 'thick' housewife to learn.

During the day I would often dust the keyboard. That may sound rather boring, but at least I had no one telling me that it wasn't my turn, or that they wanted to show me something.

I noticed that the keys were positioned in exactly the same way as the typewriter I had tried so hard



to master when I was at college. So I spent the rest of the week pretending to type 'Now is the time for all good men to come to the aid of the party', the only thing I had ever managed to type at speed.

While having breakfast a week later it dawned on me that everyone in the house knew how to work the computer but me. They had such bags around their eyes from programming all night long that they looked like a family of pandas. It was time I did something to learn.

I thought about this computer problem a great deal, and it wasn't until I was saying goodbye to the girls at the school gate that I decided what had to be done. By 9.15, I was back at home sitting in front of Brains, saying in a very determined voice: 'To hell with the washing up, cleaning, cooking, shopping, ironing and other such mundane chores'.

I was going to do something completely different today. I was going to teach myself how to use this dreaded computer. With all the confidence in the world, I cracked my fingers like a cinema safe-breaker, switched on the plug sockets, and turned on the television.

Then I hit snag number one. The screen was a mass of fuzz, with matching sound. I realised my task wouldn't be easy, so I did what every other stay-at-home housewife would and pressed all the buttons on the keyboard. Surprisingly, nothing happened. Being the level-headed, quick-thinking person that I am, I decided to read the instruction manual.

I had a slight problem here because all computer books look the same. So I started at the top of the pile and worked my way through the 20 or so books littering the floor around the computer. The first time through I had no success at all. I was expecting the book to have printed on the cover: 'I am the book you are looking for'. It didn't so I went through the pile again, more thoroughly, and hey presto: *User Guide*! I looked through the index and realised this was my kind of book. It even had a section on how to unpack it properly. But as it had already been unpacked I didn't read that section, but I made a note of the page number in case we ever bought another one.

The next section was called 'Getting going' and half way down the page I learned of the on/off switch on the back of the computer. I pressed it with some care and completely defuzzed the screen.

The I hit snag number two. What in the world did BBC Computer 32K mean? At this point I decided to stop to have a cup of coffee as it was now 10.30, the house was still a pigsty and all I had done was to switch the silly thing on.

Over coffee the guide began to give up its mysteries to me. I learned that any of the keys could be pressed while the computer was turned on without fear of my being struck by lightning. Things could be rubbed out by pressing the delete button, an important feature when the typist is me.

I spent the rest of the morning filling the screen with, you've guessed it, 'Now is the time for all good men to come to the aid of the party'. I had not lost my touch.

'What in the world did BBC Computer 32K mean?'

After a short break to flit through the house with the old vacuum cleaner and feed the starving children home for their lunch, I returned to the more interesting subject of the day.

Switching on was now a mere formality, and once again I was face-to-face with BBC Computer 32K. I wanted to load the *Welcome Tape* as I knew it held some nice programs illustrating the capabilities of this machine.

Following the instructions in the book I tried to get it to print on the screen. 'Adjust your volume control until a message is printed on the screen telling you that the setting is correct', said the guide. But I couldn't get it. I tried and tried but with no success and it was at this point that I realised I really did need the help of someone more qualified in operating computers.

I decided to, sadly, call it a day. But instead of admitting defeat to the others I would revel in my achievements.I would make myself a badge to tell the world that I switched on the BBC micro and lived. That way, if my husband had any heart, he would fall over himself to teach me the things he knew.

I went to pick the girls up from school and all my friends gathered round, intrigued by my badge. But I wasn't telling them anything about playing on computers all day and not having made the beds yet.

I approached the girls to ask them if they could help me learn in secret, then surprise Daddy at the end of the week. No, they said, they were too busy playing with their friends to stop and teach me anything. So it would have to be the great master himself.

The minute my husband arrived home from work the girls spilt the beans on the silly badge I was wearing and how I had tried to pursuade them to give me lessons in secret. So I threw the badge at him, followed by the burdens of my day.

He smiled a few times but never once called me a silly girl or any of those other things that loving husbands call their wives. When I finally wound up with 'and I haven't even had time to make the beds', he took me by the hand, turned on the computer, replaced a wire that he assured me was plugged in when he went to work that morning, and as if by magic there was the message I had nearly torn my hair out for earlier. Then he returned my slightly crumpled badge to its rightful place, and we went to eat dinner.

Things have definitely changed now. I have my own time on the computer in the evenings, with personal tuition from my husband. This enables me to practise during the day what I have learned the night before. And I still have time to make the beds!

I can take part in breakfast-time discussions. I feel part of the family once again. I read computer books in bed now instead of Catherine Cookson.

I have to admit that I do find some of them rather boring, but it makes living with a computer fanatic and his protégés that much easier.

10 101 001
00010 100
10001 101
00000000
00010111

Machine talk

▶continued from page 14

language programs, comments are essential. They help ensure that the programmer can return to a program later and understand what was designed.

In the program of figure 2 we might incorporate the following comments:

```
100 REM Assembly language program
110 REM sub 32134 on Model B for 15750
120 REM Put an 'A' on the screen
130 CLS
140 P%=&1500
150 [
160 LDA #65       \ ASCII for A
170 STA 15750     \ screen location
180 RTS           \ return to BASIC
190 ]
200 END
```

In a Basic program we often represent numeric values by giving them a variable name – more properly, we would say that we use named variables to which we assign specific values.

Labels can be used in assembly language programs to perform a corresponding function. In our simple program, a label might be used to refer to the screen location to which we shall write 'A'. This also provides a tidier solution to the problem of referring to the different locations required for the example to run on model A and model B machines. The program could then become:

```
100 REM Assembly language program
110 REM sub 32134 on Model B for 15750
115 SLOC=15750
120 REM Put an 'A' on the screen
130 CLS
140 P%=&1500
150 [
160 LDA #65       \ ASCII for A
170 STA SLOC      \ screen location
180 RTS           \ return to BASIC
190 ]
200 END
```

There are many more features of the assembler that are important to the assembly language programmer. Next month we will look at other features of the assembler, including positioning your program in memory.

INDEX OF ADVERTISERS

BIGGEST AND BEST

We're best because we're biggest. The biggest Acorn Distributor you can get.
We've got the biggest range of hardware, software, add-ons, books, stationery, peripherals, monitors and printers. And the most efficient distribution facilities in the country.

Don't try the rest—come to the best!



230/6, Derby Road, Stapleford, Nottingham NG9 7BL Tel: Nottingham (0602) 396976


PROGRAMPOWERMICROPOWER
NEW-DEVASTATING MODEL B (or 32K MODEL A) GAME FROM BRITAINS LEADING SOFTWARE HOUSE!
BBC
ALIEN DESTROYERS
MICRO POWER
Sensational, high speed program incorporating brilliant use of colour graphics and sound. 48 strong alien fleet of 3 different types plus Mothership scoring mystery bonus. Choice of 6 alien speeds and 3 bomb speeds. Vertical, angled and exploding missiles. Options to replace defences and suppress new fleet advances. Bonus bases awarded each new sheet. Scoring according to overall difficulty level plus end of game ranking of top scores.
This program also has many unique extras including "Battle Analysis" showing numbers of aliens shot down, how many Motherships destroyed, number of sheets cleared, shots fired, percentage hits and bases lost.
If you needed a reason to upgrade to 32K, now you have it!!!
ONLY £6.95
OTHER BBC PROGRAMS
MODEL A or B
CHESS (32K). Our excellent machine code program—now with superb MODE 1 graphics. Six skill levels, play back or white, illegal moves rejected, 'en passant', castling, 'take-back' of moves, and display of player's cumulative move time. Options include Blitz Chess, where you must move in 10 seconds, set-up of positions for analysis, replay of a game just played and saving of part completed game on tape. On loading a 1972 Spassky/Fisher game can be replayed. £6.95. Model A version still only £4.95. Graphics Upgrade exchange. £2.50.
ADVENTURE. All the excitement, intrigue and frustration of a main-frame adventure. Explore the tortuous forests, dark caverns and castle dungeons. Beware the maze of twisting tunnels and the desert wastelands. Outwit the predators. Rescue the Princess and carry off the treasures. Great skill and imagination are required to play this excellent game. You may never exhaust the possibilities. YES—this is our highly acclaimed ADVENTURE for the Atom!!
Other NEW 16K Titles: £8.95 FILER (Database). £5.95 each—ASTEROID BELT, SEEK (Adventure), Junior Maths Pack, Where (UK Geography). £4.95 Reversi I, £3.95 Mastermind (2-way).
MODEL B (or 32K Model A)
GALACTIC COMMANDER (Aptitude Test). Nine phase test for aspiring Space Vehicle Commanders. Begin by separating your craft from the command module and landing under acceptable gravity conditions. Then GRADUATE through combinations of lower gravities, circulating command modules, defended planets (some with homing missiles), and eventually to underground landing sites in cave systems with unfriendly inhabitants. The program presents a real challenge. The use of machine code and high-res. graphics give beautifully smooth action. Very good sound £6.95.
FOOTER. Exciting Advance in Computer Games. This is a TWO PLAYER football match, masterfully presented in HIGH-RES graphics. Out-run and out-dribble your opponent and finally blast the ball into the net. You will be amazed at the realism of this program! Newly designed graphics handling routines give a totally smooth movement to both player and ball. (See the players' legs in running action!) £6.95.
Other NEW 32K Titles: £6.95 each—METEOR BATTLE, LASER COMMAND. £5.95 each—MARTIANS, CONSTELLATION. £4.95 each—REVERSI II, MAZE INVADERS, ROULETTE. 'New Prices'—TIMETREK £6.95, ELDORADO GOLD £5.95.
All programs now available from good dealers.
We pay 20% Royalties on BBC, Spectrum, Dragon and Atom programs.
AUTHORISED BBC, ACORN AND DRAGON DEALERS
Special Offer—Deduct £1 per cassette when ordering two or more.
PLEASE NOTE!
We now offer over 30 BBC and 60 Atom programs of the highest quality and at highly competitive prices. Order NOW or send for our full catalogue.
WE Guarantee THAT ALL OUR ADVERTISED PROGRAMS HAVE BEEN COMPLETED AND ARE READILY AVAILABLE
Please add 55p order P & P + VAT at 15%
VISA
MICRO POWER LTD.
8/8A REGENT STREET
CHAPEL ALLERTON
LEEDS LS7 4PE.
Tel. [0532] 683186
PROGRAMPOWERMICROPOWER